

PROJECT 3

Yipppeeee!



CODEJIKA.COM

PROJECT 3

Training 1

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

Desktop - v1

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

Build an awesome website for a [Small Business](#) with a simple JavaScript order form.



LEARN:


The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 2.5 Hours	
3. Simple Biz Website // 5 Sessions 4 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

Diagram description: A box on the right contains a certificate icon and the text '<Certificate> Milestone Project'. Two arrows point from this box to the 'Simple Biz Website' and 'Basic Business Website' rows in the table above.

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info

2. Learn Flex, add “Cards” & content.

3. Build tables & add a form.

4. **JS** | Send & receive information from HTML.

5. **JS** | Add input from form and play hide & seek.

6. Footer, TEAM section & content writing.

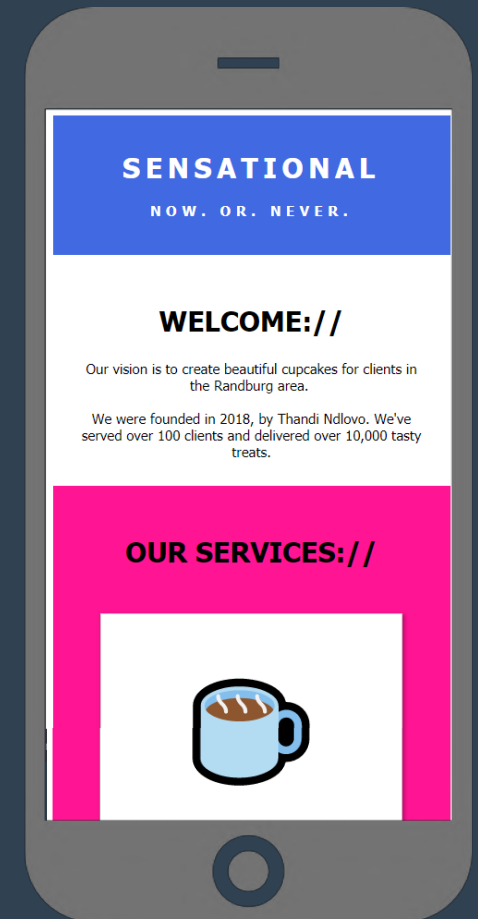
TRAINING SNAPSHOT

These are your missions for today.

1. Code the **site structure**
2. Add a **header**
3. Formulate an “Intro” section.

PROJECT 3 – TRAINING 1

PREVIEW



MISSION

Code the *site structure*.

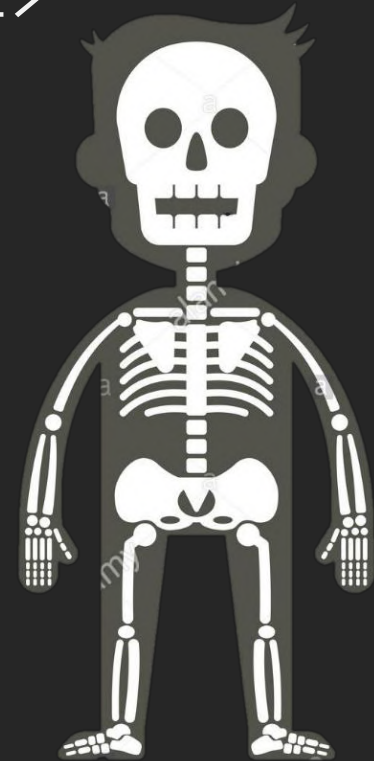


This mission will include 2 challenges.

START.

Start by setting up the skeleton of your code.

```
<!DOCTYPE html>  
<head>  
  <style>  
  </style>  
</head>  
<body>  
  
</body>
```



CHALLENGE

STEP 1

TIP:

Save a new notepad file as `.html`.

CODE:

Add this section:

```
<!DOCTYPE html>
<head>
  <style>
  </style>
</head>
```


CHALLENGE

STEP 2

CODE:

Now add `<header>` and `<section>` in `<body>`.

```
<body>  
  <header>  
  </header>  
  <section>  
  </section>  
</body>
```

GREAT JOB!



THINK

What's the name of your
business?



Or who are you making it
for?

MISSION

Add a simple header, h1 & sub-title.



This mission will include 5 challenges.

CHALLENGE

STEP 1

CODE:

Add an `<h1>` in `<header>` with your company name or brand.

```
<h1>  
  Awesome Industries Inc.  
</h1>
```

CHALLENGE

STEP 2

CODE:

Under the `<h1>` place an `<h4>` with the company tagline.

```
<h4>  
    Cute Cupcakes for Best  
    Besties  
</h4>
```

NICE.

Keep it up.



Now let's style the
<body> and
<header>.

CHALLENGE

STEP 3

TIP:

```
body { }
```

CODE:

Add a body selector in CSS with these styles:

```
font-family: tahoma;  
text-align: center;
```

For your Company name
make it **white** and have the
letters **spaced-out**.

CHALLENGE

STEP 4

CODE:

Add `header { }` with a background color and some padding:

```
letter-spacing: 6px;  
background: royalblue;  
padding: 20px;
```

CHALLENGE

STEP 5

TIP:

`color: ???;`

CODE:

Make the text in header { }
white with a CSS rule.

CHECKPOINT

Your code should look like this:

```
<!DOCTYPE html>
<head>
  <style>
    body {
      font-family: tahoma;
      text-align: center;
    }
    header {
      letter-spacing: 6px;
      color: white;
      background: royalblue;
      padding: 20px;
    }
  </style>
</head>
```

MARVELOUS!

Keep it up.



MISSION

Create an “Intro” section with content about the business.



This mission will include 9 challenges.

ALRIGHTY THEN.

1. Stick a sub-title "Welcome" in a `<h2>`.
2. Explain the business in a `<p>`.

CHALLENGE

STEP 1

CODE:

Add an `<h2>` with this text in it:

WELCOME:

WHERE:

Inside `<section>`.

NEXT.

Tell visitors of the site
about the vision of the business.

CHALLENGE

STEP 2

CONTENT:

Write a short Vision Statement.

Here is an example:

VISION: To create beautiful cupcakes for clients in the Randburg area.

CHALLENGE

STEP 3

CODE:

Insert your vision statement in
a `<p>`.

WHERE:

After the welcome `</h2>` closing tag
and still inside section.

YOU'RE GETTING THERE.

Keep it up.

CHALLENGE

STEP 4

CODE:

Insert 2 `
` tags.

WHERE:

After the VISION section. After the closing tag `</p>`.

NOW.

Create an “About Us” section.

CHALLENGE

STEP 5

CONTENT:

Create an “About Us” explanation.

Here is an idea:

ABOUT US: Founded in 2018, by Thandi Ndlovo.

We've served over 100 clients and delivered over 10,000 tasty treats.

CHALLENGE

STEP 6

CODE:

Add the “About Us” content in
a `<p>`.

WHERE:

After the VISION and the 2 `
`s.

CHALLENGE

STEP 7

CODE:

Style h2 with `font-size: 2em;` and `width 100%.`

NOTE:

The `width: 100%;` rule will make sure that the text does not wrap around other content.

CHALLENGE

STEP 8

TIP:

```
section { }
```

CODE:

```
Give the <section>s 30px  
padding and margin-bottom:  
40px; .
```

THAT'S AMAZING!

You just added **35 lines of code**
in the first "Training".



CHECKPOINT

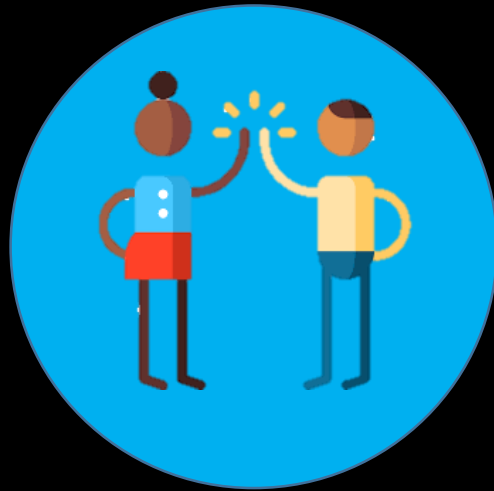
Your code should look like this:

NOTE:
Content should be
your own.

```
    h2 {
      font-size: 2em;
      width: 100%;
    }
    section {
      padding: 30px;
      margin-bottom: 40px;
    }
  </style>
</head>
<body>
  <header>
    <h1>
      Awesome Industries Inc.
    </h1>
    <h4>
      Cute Cupcakes for Best Besties
    </h4>
  </header>
```



CONGRATULATIONS



You've finished
Training 1.

YOU'RE DONE.



Training 2 is waiting for you.

WANT THE LATEST NEWS AND BONUS SECTIONS?

Sign-up at: www.CodeJIKa.com



LOADING . . .



CODEJIKA.COM

PROJECT 3

Training 2

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

Desktop - v1

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

Build an awesome website for a [Small Business](#) with a simple JavaScript order form.



LEARN:


The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 2.5 Hours	
3. Simple Biz Website // 5 Sessions 4 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

Diagram description: A box on the right side of the table, containing a certificate icon and the text '<Certificate> Milestone Project'. Two arrows point from this box to the '3. Simple Biz Website' and '6. Basic Business Website' rows of the table.

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info

2. Learn Flex, add "Cards" & content.

3. Build tables & add a form.

4. **JS** | Send & receive information from HTML.

5. **JS** | Add input from form and play hide & seek.

6. Footer, TEAM section & content writing.

DUMELA*.



*"Hello" in Sotho.

TRAINING SNAPSHOT

PROJECT 3 – TRAINING 2

These are your missions for today:

1. Learn about CSS Flex.
2. Use CSS Flex to make cool looking boxes for your content.

MISSION

Learn about CSS Flex



This mission will include 3 challenges.

Here we go.

Flexbox

Flex allows you to easily space boxes or images.

DESKTOP:



Example: For an image gallery.

Flexbox

Flexbox

And they will shrink or stack based on the screen width.

DESKTOP:



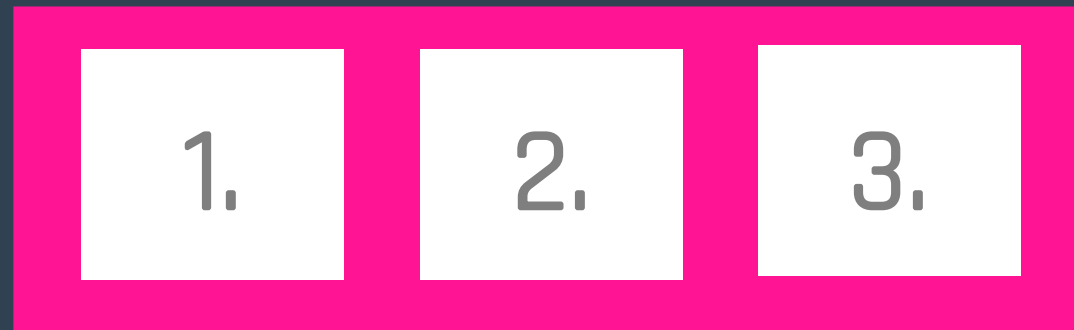
MOBILE:



Flexbox

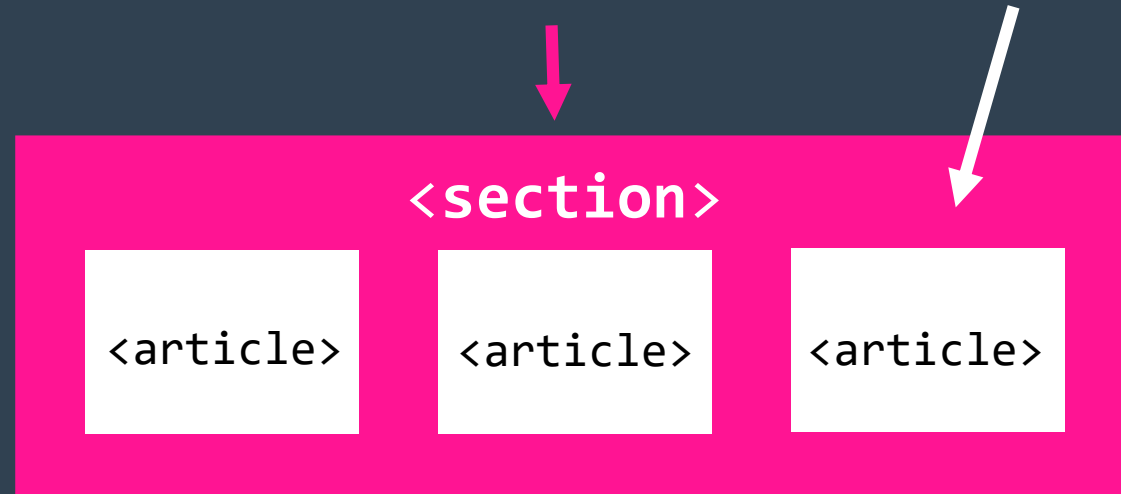
To use flex, start with

1. A **container**.
2. Then add your **content**.



Flexbox | Example

STEP 1: Add the **container** and **content**.



```
<section class="container">  
  <article> </article>  
  <article> </article>  
  <article> </article>  
</section>
```

Flexbox | Example

STEP 2: Add the "Flexbox" Magic.

Flexbox

Container CSS



Flexbox Magic



```
.container {  
  display: flex;  
  justify-content: center;  
}
```

Flexbox | Example

STEP 3: Tell "Flexbox" how to arrange the content.

1. Container CSS



2. Flexbox Magic





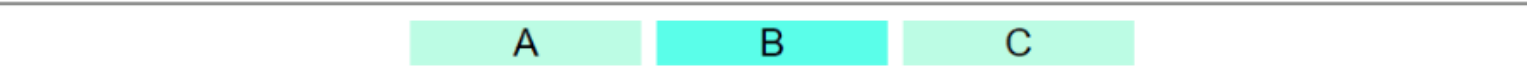
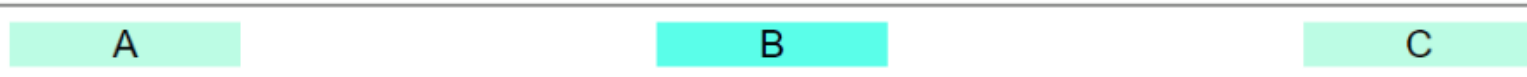
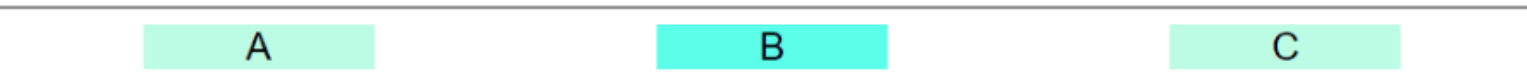

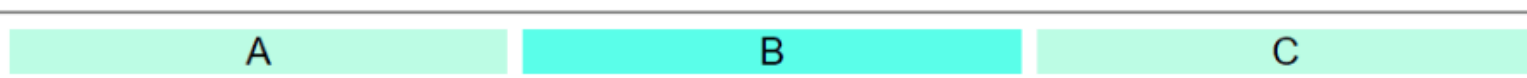
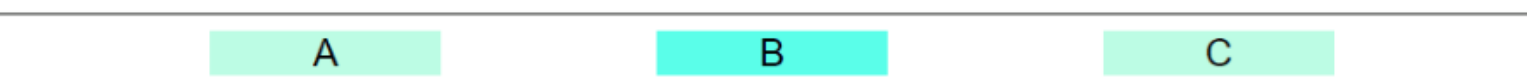
3. How to Arrange:



```
.container {  
  display: flex;  
  justify-content: center;  
}
```

CSS FLEX REFERENCE:

justify-content

<i><default></i> flex-start	
flex-end	
center	
space-between	
space-around #1	
space-around #2	
stretch	
space-evenly	



JavaScript Teacher

REFERENCE: https://medium.com/@js_tut

LET'S FLEX!



CHALLENGE

STEP 1

TIP:

```
.container { }
```

CODE:

Create a CSS class
`.container` with:

```
display: flex;  
justify-content: center;  
flex-wrap: wrap;
```

THAT'S FLEX-TASTIC !



CHALLENGE

STEP 2

WHERE:

Place below "Welcome"
section closing tag
`</section>`.

CODE:

Add a div styled with **container** class and 2x `<article>` tags.

```
<section class="container">  
  <article> </article>  
  <article> </article>  
</section>
```

CHALLENGE

STEP 3

CODE:

Insert a sub-title with `<h2>` at the top of the section: SERVICES://

```
<section class="container">  
  <h2>  
    SERVICES://  
  <h2>
```

SUPER



Now we're going to add a card design to the `<article>` s.

MISSION

Learn & create a “Cards” design style.



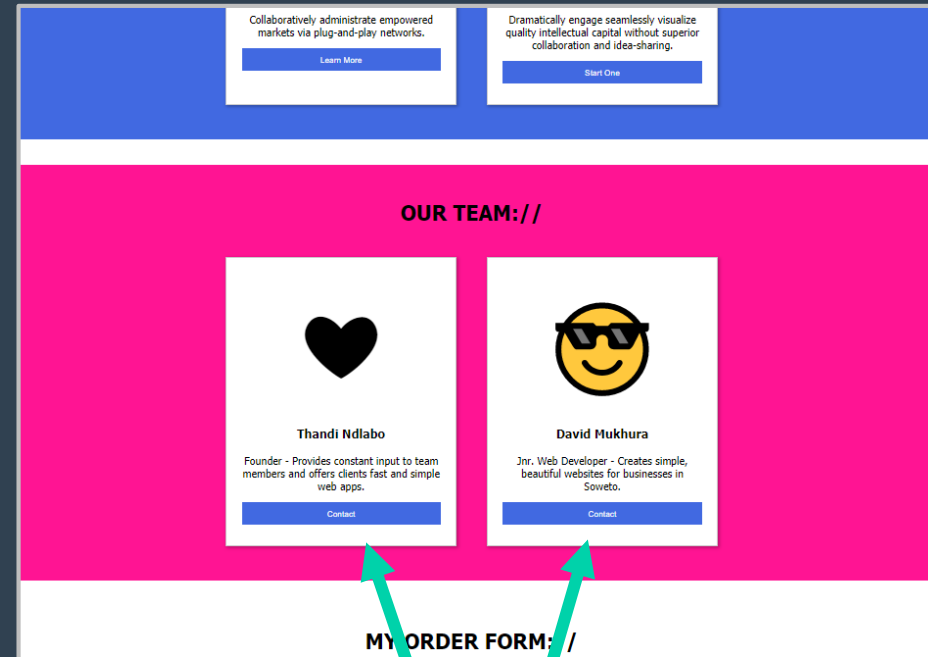
This mission will include 5 challenges.

CARDS

Your flex content will look like "cards".

This is a popular design style.

DESIGN STYLE: CARDS

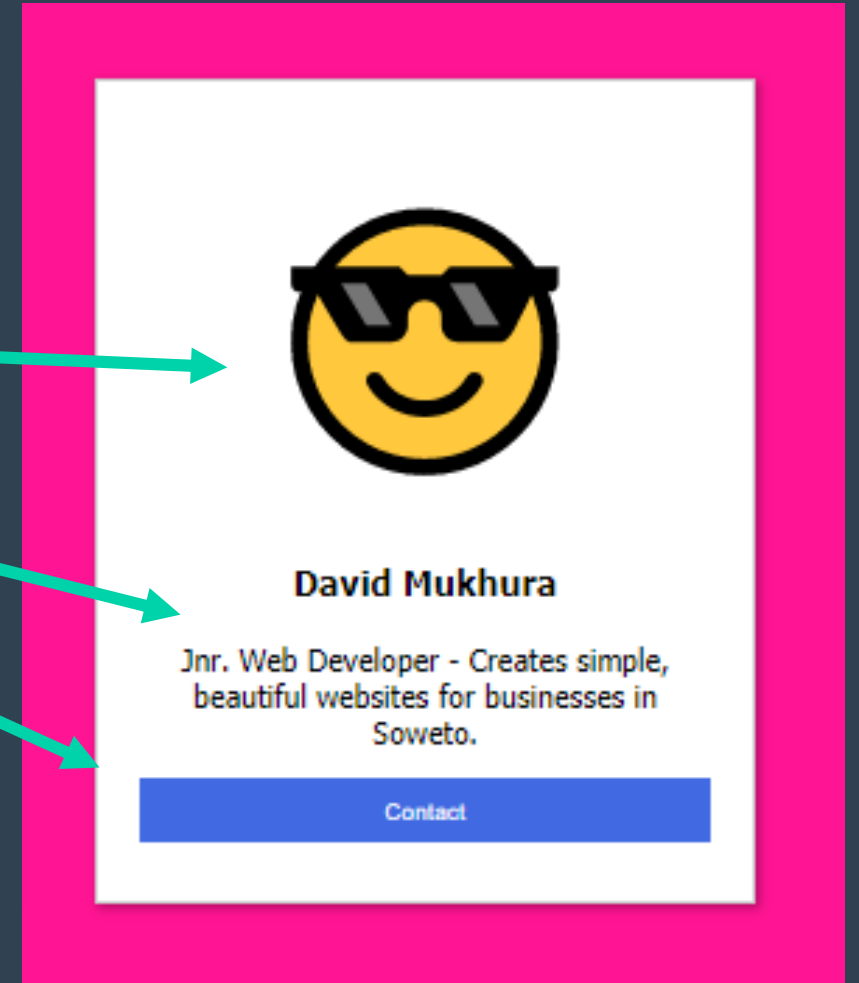


CARDS
<article>

CARDS

Each “Card” has three elements inside:

1. Icon
(or image)
2. Text
3. Button



CARDS

Your code will look like this:

```
<article class="card">
```

1.

```
<div class="icons"> &#x1F60E; </div>
```

2.

```
<h3>David Mukhura</h3>  
<p>Jnr. Web Developer - Creates beautiful  
websites for businesses in Soweto.</p>
```

3.

```
<button> Contact </button>
```

```
</article>
```

ANDIAMO*!



*ITALIAN: "Let's do this!"

CHALLENGE

STEP 1

TIP:

```
<article class="card">
```

CODE:

Create a CSS class called `.card` and link to it in each `<article>` tag.

```
.card {  
  border: 1px solid #ccc;  
  background-color: ivory;  
}
```

CHALLENGE

STEP 2

CODE:

Add a padding and margin to the `.card` class.

```
.card {  
  border: 1px solid #ccc;  
  background-color: ivory;  
  margin: 25px;  
  padding: 25px;  
}
```

CHALLENGE

STEP 3

TIP:

You can put this at the end of the CSS code.

CODE:

Adapt flex and the “Cards” to show nicely on larger screens.

Add this CSS:

```
@media screen and (min-width: 50em) {  
  .card {  
    flex-basis: 325px;  
  }  
}
```

CHALLENGE

NOT RELATED TO "CARDS."

CODE:

Add this to make the site header look better on small screens:

```
@media screen and (min-width: 50em) {  
  .card {  
    flex-basis: 325px;  
  }  
  header h1 {  
    font-size: 5em;  
  }  
}
```

FANCY



Let's add a **box-shadow** to our
`.card` .

The next 3 slides are optional.

BRIEFING

1.1.2

box-shadow

```
box-shadow: 6px 6px 6px yellow;
```

`box-shadow` is used by the experts to make cards & pictures look more elegant.

6px

1st Right

6px

2nd Bottom

6px yellow;

3rd Blur
Effect

4th Color

TIP:

Blur means to make something fuzzy.

TIP:

Transparent means "See-through."

TIP 2:

This doesn't work in all browsers.

RGBA Transparency Trick

RGBA color: Use the last number to make it semi-transparent.

`rgba (0 , 0 , 0 , 0.3)`



Red



Green



Blue

Alpha

WHAAAAT !?!?!?!?



If you feel like this, don't worry.

CHALLENGE

STEP 4

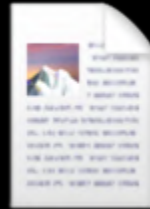
OPTIONAL:

This is an optional challenge. You can skip it if you like.

CODE:

Use both `box-shadow` and a RGBA color to make your `.card` super PRO.

```
.card {  
  border: 1px solid #ccc;  
  background-color: ivory;  
  margin: 25px;  
  padding: 25px;  
  box-shadow: 6px 6px 6px rgba(0,0,0,0.3);  
}
```



Now we're going to add
content to the cards.

MISSION

Add 3 content elements to the “Cards”.

1. **Add an icon** class & div.
2. **Write** the **content**.
3. **Insert a button** & style it.



This mission will include 7 lengthy challenges.

LET'S GO.

CHALLENGE

STEP 1

CODE:

Create a CSS class called `.icons`.

```
.icons {  
  font-size: 8em;  
  padding: 25px;  
}
```

CHALLENGE

STEP 2

CODE:

Create a `<div>` nested inside each `<article>`.

Insert an emoji in your `<div>`.

```
<article class="card">
  <div class="icons"> 🍌;
</div>
</article>
```


LEKKER*!



*AFRIKAANS: "Nice one!"

CHALLENGE

STEP 3

WHERE:

Below the icons `</div>` closing tag.

CODE:

Add a `<h3>` sub-title and a `<p>` of explainer text in each `<article>`.

EXAMPLE:

```
<h3>David Mukhura</h3>
```

```
<p>Jnr. Web Developer - Creates simple, beautiful websites for businesses in Soweto.</p>
```

CHALLENGE

STEP 4

WHERE:

Below the `</p>` closing tag.

CODE:

Add a `<button>` and closing tag `</button>` with the text "Contact" in the middle.

Do this in each `<article>`.

EXAMPLE:

```
<button> Contact </button>
```

CHALLENGE

STEP 5

WHERE:

Below the `</p>` closing tag.

CODE:

Add a `<button>` and closing tag `</button>` with the text "Contact" in the middle.

Do this in each `<article>`.

EXAMPLE:

```
<button> Contact </button>
```

Style Multiple Classes

You can style multiple classes with the same code. Just separate them with a comma.

EXAMPLE:

```
.button, .container {  
  margin: 10px;  
}
```

CHALLENGE

STEP 6

CODE:

Style the `<button>` tag and the `.button` class with this CSS:

Note: We'll use the class later.

```
button, .button {  
  background: royalblue;  
  border: 0;  
  color: white;  
}
```

CHALLENGE

STEP 7

CODE:

Add 3 rules to the button CSS:

```
button, .button {  
  background: royalblue;  
  border: 0;  
  color: white;  
  padding: 10px;  
  width: 100%;  
  margin-bottom: 10px;  
}
```

YAAAAAYYY!!!



You're done.

CHECKPOINT

Your CSS code should look like this:

```
/* CSS CARDS */
.container {
display: flex;
justify-content: center;
flex-wrap: wrap;
}

.card {
border: 1px solid #ccc;
background-color: ivory;
margin: 25px;
padding: 25px;
box-shadow: 6px 6px 6px
  rgba(0,0,0,0.3);
}
```

```
.icons {
font-size: 8em;
padding: 25px;
}

button, .button {
background: royalblue;
border: 0;
color: white;
padding: 10px;
width: 100%;
margin-bottom: 10px;
}

.blue {
background: teal;
}
```

CHECKPOINT

Your HTML code should look like this:

```
<section class="blue container">
  <h2>
    OUR SERVICES://
  </h2>
  <article class="card">
    <div class="icons">&#9749;</div>
    <h3>Sub-title</h3>
    <p> ... text. </p>
    <button>Learn More</button>
  </article>
  <article class="card">
    <div class="icons">&#9734;</div>
    <h3>Sub-title</h3>
    <p> ... text. </p>
    <button>Learn More</button>
  </article>
</section>
```

BONUS

CHALLENGE

STEP 1



CLASSY BLUE

25 POINTS

TIP:

Add a space between **blue** and **container**.

CODE:

1 of 2

Add another class to `<section>` tag:
blue

```
<section class="blue container">
```

BONUS

CHALLENGE

STEP 2



CLASSY BLUE

25 POINTS

CODE:

Add this class.

2 of 2

```
.blue {  
  background: teal;  
}
```

WELL DONE!

This was a long, hard lesson.

QUIZ

1. 2.



ANSWER: NEXT SLIDE

QUESTION 3.2.1

What is flex for?

- A. It allows you to organize content simply.
- B. It's a body building technique.
- C. Flexbox is a way of creating boxes.
- D. All of the above.

QUIZ

1. 2.



PREVIOUS ANSWER: A.

QUESTION 3.2.2

What's RGBA?

- A. A type of HTML color code.
- B. Red, green, blue & alpha.
- C. Allows adding transparency to boxes.
- D. All of the above.

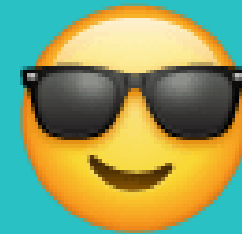
QUIZ

WHAT WAS YOUR SCORE?

1. 2.

50% 100%

PREVIOUS ANSWER: D.



SUGGESTION:

If you have internet:
Download and install a
better code editor.

WHY?

It's easier and shows any
mistakes in your code.

We recommend:
Notepad++



FOLLOW THE LINK:

<https://notepad-plus-plus.org/download/>

Ready for
Training 3?



#CODEJIKA



Help other kids learn to code.

Post on Social Media what you're
working on with #CodeJIKA .

LOADING . . .



CODEJIKA.COM

PROJECT 3

Training 3

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A LANGUAGE OF SOUTH AFRICA.

Desktop – v6

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

You'll create a website for a [Small Business](#) with an simple order form and a few lines of JavaScript.



LEARN:


The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 2.5 Hours	
3. Simple Biz Website // 5 Sessions 4 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

Diagram description: A box on the right side of the table, containing a certificate icon and the text '<Certificate> Milestone Project'. Two arrows point from this box to the '3. Simple Biz Website' and '6. Basic Business Website' rows of the table.

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info

2. Learn Flex, add “Cards” & content.

3. Build tables & add a form.

4. Automatic Order Summary with JavaScript.

5. Footer, TEAM section & content writing.

JAMBO*.



*"Hello" in Swahili.

TRAINING SNAPSHOT

PROJECT 3 - TRAINING 3

These are your missions for today:

1. **Build** tables.
2. Create a **form** from a table.
3. Add some **buttons**.

MISSION

Build tables.



This mission will include 4 challenges.

YALLA*.

*"Come on" in Arabic.

<table>

Tables consist of;

A. Columns <th>

B. Rows <tr>

Column 1	COLUMN	Column 3
Row 2		
Row 3		
ROW		

Example: HTML Table

<table>

Tables are a great way to store and show information.

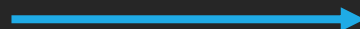
Tables can also structure content, in a web form, for example.


Name	Age	Phone
Themba	14	074 7858588
John	18	098 5211611
Rachael	17	062 3241279

Example: HTML Table

<table>

Example: HTML Table

```
<table>
  <tr>  ROW 1
    <td>Name:</td> CELL 1
    <td>Age:</td> CELL 2
    <td>Phone:</td> CELL 3
  </tr> ROW END

  <tr>  ROW 2
    <td>Themba</td> CELL 1
    <td>14</td> CELL 2
    <td>074 7858588</td> CELL 3
  </tr> ROW END

  . . .
```

<table> | Resource

Options: Table Styling:

`<th>` Table Header cell

`<td rowspan="2">` Merge two cells

`<table style="width:100%">` Style

`<table class="t01">` HTML Class

`.t01 {` CSS Class Style

`width: 100%;`

`background-color: #f1f1c1;`

`}`

LET'S BUILD TABLES!



`<table>`

Let's make a new `<section>`
with a table in it.

CHALLENGE

STEP 1

WHERE:

Below section with
"Cards."

CODE:

Create a new `<section>`
with an `<h2>` sub-title in it.

```
<section>  
  <h2> MY ORDER FORM: </h2>  
</section>
```

CHALLENGE

STEP 2

WHERE:

Below `</h2>` and
before `</section>`.

CODE:

Build this table:

```
<table>
  <tr>                                ROW 1
    <td> Name: </td>
    <td> </td>
  </tr>

  <tr>                                ROW 2
    <td> Address: </td>
    <td> </td>
  </tr>
</table>
```

CHALLENGE

STEP 3

WHERE:

After row 2 and before the `</table>` closing.

CODE:

Add two more rows:

• • •

```
<tr>                                ROW 3
  <td> Favourite drink: </td>
  <td> </td>
</tr>
```

```
<tr>                                ROW 4
  <td> Quantity: </td>
  <td> </td>
</tr>
```

• • •

BOOM!



You just built your first table.

CHALLENGE

STEP 4

WHERE:

You should know where by now. ;)

CODE:

1. Pad table cells with 10px.
2. Center `<table>`.

```
td {  
    padding: 10px;  
}
```

```
table {  
    margin: auto;  
}
```

MISSION

Add a form to a table.



This mission will include 6 challenges.

BRIEFING

1.1.2

HTML Form

Definitions:

**User : Anyone who uses your website.*

**Input : Information given.*

<form>

Form is used to collect user* input*.

GOOFY FORM:

What's the silliest idea you have ever had?

Example: HTML Form

<form>

Example: Simple HTML Form

```
<form>
  <h1>GOOFY FORM:</h1>
  What's the silliest idea you have ever had?
  <br>
  <input type="text" size="50">          INPUT 1
  <br><br>
  <button>Submit</button>             SUBMIT BUTTON
</form>
```

```
<form action="">
```

Example: Form Actions

```
<form
```

```
  id="my-form"
```

CSS ID styling #my-form

```
  action="me.html"
```

Where data is sent

```
  target="_blank"
```

_blank = new tab

```
>
```

_self = same page

CHALLENGE

STEP 1

CODE:

Wrap the whole table in `<form>` tags.

- With the `id="my-form"`.

```
<form id="my-form" >  
    ... table ...  
</form>
```

`<input>`

Example: Form Inputs

`<input type="text" → Types: Text, radio & submit.`
`size="50" → Width input field in px.`
`name="idea" → Name for saving information.`
`value="Idea..." > → Placeholder text.`

REFERENCE

1.1.2

HTML INPUT TYPES

More online here:

https://www.w3schools.com/html/html_form_input_types.asp

HTML INPUT TYPES

SET 1

TYPES

```
<input type="button">
```

```
<input type="checkbox">
```

```
<input type="color">
```

```
<input type="date">
```

```
<input type="datetime-local">
```

```
<input type="email">
```

```
<input type="file">
```

```
<input type="hidden">
```

```
<input type="image">
```

```
<input type="month">
```

SET 2

TYPES

```
<input type="password">
```

```
<input type="radio">
```

```
<input type="range">
```

```
<input type="reset">
```

```
<input type="search">
```

```
<input type="submit">
```

```
<input type="tel">
```

```
<input type="text">
```

```
<input type="time">
```

```
<input type="url">
```

```
<input type="week">
```

CHALLENGE

STEP 2

CODE:

Insert an `<input>` in the second cell (`<td>`) of the first row.

```
<tr>
  <td>Name:</td>
  <td><input type="text" size="25"
name="my-name"></td>
</tr>
```

CHALLENGE

STEP 3

CODE:

Insert an `<input>` in the second cell (`<td>`) of the second row.

```
<tr>
  <td>Address:</td>
  <td><input type="text" size="25"
name="my-address"></td>
</tr>
```

AMAZING!



You are “Acing” this `<form>` thing.

CHALLENGE

STEP 4

CODE:

Insert a `<select>` option in the second cell (`<td>`) of the third row.

```
<tr>
  <td>Favourite drink:</td>
  <td>
    <select name="my-drink">
      <option> Milk </option>
      <option> Coffee </option>
      <option> Tea </option>
    </select>
  </td>
</tr>
```

BRIEFING

1.1.2

Form Type Number

```
<form type="number">
```

type="number" allows to limit the numbers the user can choose.

Example: Form Type Number

```
<input type="number" name="my-qty"  
value="1" min="1" max="5">
```

↓
DEFAULT*
VALUE

↓
MINIMUM
VALUE

↓
MAXIMUM
VALUE

DEFINITION:

**Default : The standard. This value remains in effect unless changes are made.*

CHALLENGE

STEP 5

CODE:

In ROW 4, add a “number” type of `<input>` .

Define a minimum of 1 and a maximum of 5.

```
<tr>
  <td>Quantity:</td>
  <td>
    <input type="number" name="my-qty"
      value="1" min="1" max="5">
  </td>
</tr>
```

CHALLENGE

STEP 6

CODE:

Add a short note below the `<input>` to notify the user of the limits.

Make it `<small>` with HTML.

```
<td>Quantity:</td>
<td>
  <input type="number" name="my-qty"
  value="1" min="1" max="5">
  <small>(max 5)</small>
</td>
```

GOOD JOB!



You've just added an awesome form to
the table.

Almost done.



Now just add a “Submit”
button.

MISSION

Add some buttons to the form.



This mission will include 6 challenges.

CHALLENGE

STEP 1

CODE:

Add two rows.

- We need a bit of space, so add a `
` in the fifth ROW (below the "Quantity:" ROW.)

```
</tr>  
<tr>                                ROW 5  
    <br>  
</tr>  
  
<tr>                                ROW 6  
</tr>  
</table>
```


CHALLENGE

STEP 2

CODE:

Merge two cells in the 6 ROW together using: `<td colspan="2" >`

```
</tr>
<tr>                                ROW 5
  <br>
</tr>
<tr>                                ROW 6
  <td colspan="2" >
  </td>
</tr>
</table>
```

CHALLENGE

STEP 3

CODE:

Add a button in ROW 6 using `type="button"` and a value of "Process Order".

```
<tr>                                ROW 6
  <td colspan="2" >
    <input
      type="button"
      value="Process Order" >
    </td>
</tr>
```

CHALLENGE

STEP 4

CODE:

Add a second button with `type="reset"` and a value of "Clear".

```
<tr>
  <td colspan="2" >
    <input
      type="button"
      value="Process Order" >
    <input type="reset" value="Clear" >
  </td>
</tr>
```

DO YOU REMEMBER?

In Training 2, you styled:

```
button, .button {...
```

You'll link to that styling in the next Challenge.

CHALLENGE

STEP 5

WHERE:

Within the `<input>` tags themselves.

CODE:

Add styling to BOTH the buttons using `class="button"` .

```
<td colspan="2" >  
  <input class="button"  
    type="button"  
    value="Process Order" >  
  <input class="button"  
    type="reset"  
    value="Clear" >  
</td>
```

ROW 6

CHECK

To make sure your buttons
are reflecting the styling
correctly.

DEBUG

COMPARE YOUR CODE

There should be a HTML file in the folder with all the code for Training 1 -3.

File: `CodeJIKA_Project3_T3_v191.html`

Go and check there if your code has some problems and that haven't figured out yet.

GUESS WHAT???





You've written over **150 lines**
of code in this PROJECT.



We think that's something
to celebrate about.



Show your friends
what your site looks like.



Use: `#codejika`
So that we can see too.



QUIZ

1. 2.



ANSWER: NEXT SLIDE

QUESTION 2.1.1

What is input action for?

- A. Allows the form to save submitted content.
- B. Uses `_blank` to clear the form when done.
- C. Says where to show the results of the form.
- D. All of the above.

QUIZ

1. 2.



PREVIOUS ANSWER: C.

QUESTION 2.1.2

What are input types?

A. Radio, number, text, date & much more.

B. Variable, type, style, action

C. Text, number, value, button

D. Value, button, submit, text & reset.

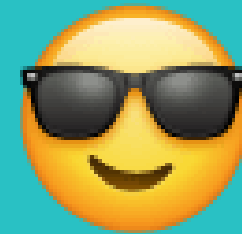
QUIZ

WHAT WAS YOUR SCORE?

1. 2.

50% 100%

PREVIOUS ANSWER: A.



It's time for Training 4
and some JavaScript.

Are you ready?



LOADING . . .



CODEJIKA.COM

PROJECT 3

Training 4

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

Desktop - v18 2019

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

Build an awesome website for a [Small Business](#) with a simple JavaScript order form.

LEARN:

The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)



#CODEJIKA




Help other kids learn to code.

Post on Social Media what you're
working on with #CodeJIKA .

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 3 Hours	
3. Simple Biz Website // 6 Sessions 4.5 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info
2. Learn Flex, add “Cards” & content.
3. Build tables & add a form.
4. **JS I** Send & receive information from HTML.
5. **JS I** Add input from form and play hide & seek.
6. Footer, TEAM section & content writing.

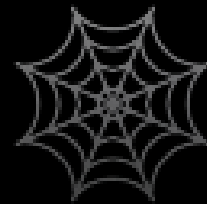
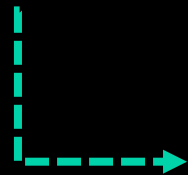
MORNING.



So glad you're here today.

WHAT'S JavaScript?

JavaScript is the **programming language** of HTML and the **Web**.



REMEMBER:

What's **JavaScript**?

CaPiTaLs matter.

If you get one "CAPITAL" letter wrong
ALL your code might not work.



JavaScript = JS

Going forward, we'll
just call it "JS".



It's shorter and easier.

WHAT IF I DON'T UNDERSTAND?



Don't worry.
Today is just a **quick intro**.

JS will be **explained more** in PROJECT 4.

TRAINING SNAPSHOT

PROJECT 3 - TRAINING 4

Missions for today.

Create an **automatic** order **summary**:

1. Link a HTML **Event** & JS **Function**.
2. Create the **order summary section**.
3. **Send results**: Variable & HTML Method.

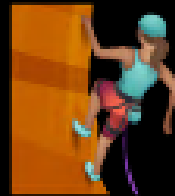
MISSION

Link a HTML **Event** & JS **Function**.

• • •

This mission will include 4 challenges.

JavaScript



Let's go!!!

THIS IS
IMPORTANT:



JavaScript | `<script>`

JavaScript is written between `<script>` tags.

```
<script>
```

```
JavaScript Code
```

```
</script>
```

WHERE:

At the end of site before `</body>` closing tag.

CHALLENGE

STEP 1

CODE:

Add script tags before the closing `</body>` tag.

```
<script>
```

```
</script>
```

```
</body>
```

GOOD!

HTML Events

Connect user **actions** on the website **to JavaScript** (& other things.)



An HTML Event is where JS comes to make connections (friends).

HTML Event TYPES

HTML EVENT TYPES

onclick	Fires on a mouse click on the element
onkeypress	Fires when a user presses a key
onmousemove	Fires when the mouse pointer is moving while it is over an element
onload	Fires after the page is finished loading
onoffline	Script to be run when the browser starts to work offline
onresize	Fires when the browser window is resized
oninput	Script to be run when an element gets user input

Note: these are just a few. There are many more.

More online here:

https://www.w3schools.com/tags/ref_eventattributes.asp

`onclick=""`

When **mouse** is **clicked**.



When someone clicks this button (or image) then run this JavaScript function.

A **HTML Event** will **trigger** a JS **Function**



...when button is clicked.

HTML Event: onclick

1.



2.

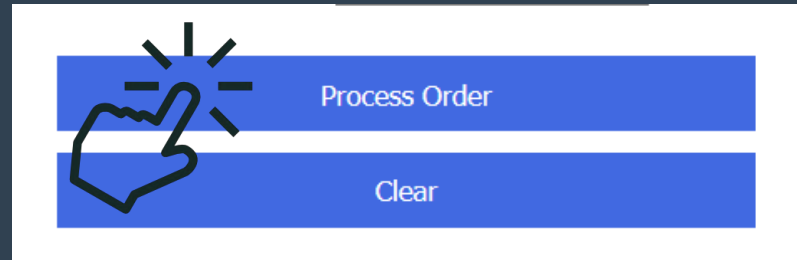


3.



onclick=""

3-STEP EXAMPLE:



```
<input class="button"
  type="button"
  value="Process Order"
  onclick="placeOrder();" >
```

```
<script>
  function placeOrder() {
  }
</script>
```

onclick event
in button.

Triggers
function `placeOrder()`.

Function `placeOrder()`
is **run**.

CHALLENGE

STEP 2

CODE:

Add the `onClick` HTML Event to the "Process Order" `<input>` with `placeOrder()` function.

```
<input class="button"  
  type="button"  
  value="Process Order"  
  onClick="placeOrder();" >
```

RAD.

Let's learn `function`.



FUNCTION

IT WORKS LIKE THIS:



JS | function

It makes **things happen**.

Name



```
function placeOrder () { }
```



Calculations go in here.

REMEMBER

What's a **function**?

CHALLENGE

STEP 3

WHERE:

Inside `<script>` tags.

CODE:

Add a function called
“`placeOrder ()`”.

```
<script>  
    function placeOrder () {  
    }  
</script>
```

TIP:

JS:



`placeOrder` (With a CAPITAL "O".)

CHALLENGE

STEP 4

CODE:

Add a success message inside the function using:

```
results = "<h3>Success</h3>...";
```

```
function placeOrder () {  
  results = "<h3>Success!</h3> Here is your order.";  
}
```

TIP:



JS:

Put a **;** semi-colon
at the end of each line.

GOOD JOB.



You just finished mission 1.

MISSION

Create the order **summary section**.



This mission will include 2 challenges.

LINK THE MASTERS

CSS



An **ID**

HTML



to a **<div>**

JS



& a **var.**

#my-order

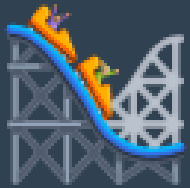
STRUCTURE:

Function

uses

Variable 2

CSS ID



`<div ID>`



Styling.

...to send the results to div.

CSS | id

Like a CSS class, just with a # instead of . (dot.)

```
CSS    #my-order { }
```

```
HTML  <div id="my-order">  
      </div>
```

CHALLENGE

STEP 1

WHERE:

Of course, in the CSS `<style>` section.

CODE:

Create a CSS ID called “my-order” with this styling:

```
#my-order {  
  background-color: #29C1C4;  
  padding: 25px;  
}
```

CHALLENGE

STEP 2

WHERE:

Immediately before the `</form>` closing tag.

CODE:

Create a `<div>` with an "my-order" id inside.

```
        </tr>
    </table>
    <br>
    <div id="my-order">
    </div>
</form>
```

ALRIGHTY THEN.



What's a variable?

MISSION

Send results: Variable & HTML Methods.

• • •

This mission will include 3 challenges.

VARIABLE

JS | var

A vehicle to **store data** and **give commands**.



STORE DATA:

“Yaay. I love learning.”

GIVE COMMANDS:

“Get me some food.”

VARIABLE

IT WORKS LIKE THIS:



JS | var

Every variable starts with **var** and has a name.

Variable

Data to store

```
var store = "I like cats."
```

Variable Name

VARIABLE

IT WORKS LIKE THIS:

JS | var

Variables can store data from tons of places.

Variable

Data to store

The diagram shows a code snippet `var store = input from all sorts of sources...` on a dark background. Three arrows point to parts of the code: a pink arrow points to `var`, a teal arrow points to `store`, and a white arrow points to the text `input from all sorts of sources...`. Labels are placed above and below the code to identify these parts.

```
var store = input from all sorts of sources...
```

Variable name



Variables can also store data typed in a form.

Why would I want to store that?



Because maybe we want to show it later in a different format or send it to someone.

CHALLENGE

STEP 1

CODE:

Add a JS variable called `orderResults` in the function.

```
function placeOrder() {  
  results = "<h3>Success!</h3> Here is your order.";  
  var orderResults =  
}
```

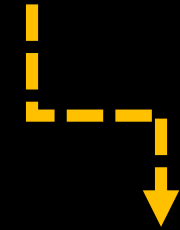
REMEMBER

What's a **variable**?

Variable: To send or **pull data** I need two things;

1. **HTML Method** to pull data.

2. Say from where (what **ID**).



```
var orderResults = document.getElementById("my-order");
```

HTML Methods

Connect JS data and results
with HTML.



HTML Methods are Heros with **super-powers** that help all the parts of the site talk to each other.

HEROS WITH SUPER-POWERS.

Variable 2



asks

gEBI



INNER



`</div ID>`



A. Link to the **ID** name.

B. Show the results **in HTML**.

.getElementById()

HTML Method: gEBI

get
Element
By
Id



Hi, my name is

gEBI.

I am a HTML
Method.

MY SUPER-POWER:

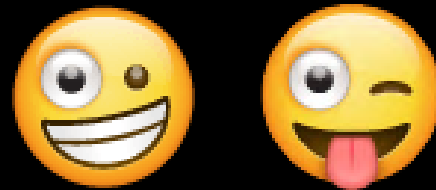
I **create a link-up** to send or pull information from anywhere, on a webpage or the internet.

FOR EXAMPLE:

Link variable to `<div>` with the `id="my-order"`.

`document.getElementById("my-order");`

READY?



CHALLENGE

STEP 2

CODE:

Add **gEBI** (`getElementById`) to send results to ID: "my-order".

```
function placeOrder() {  
  results = "<h3>Success!</h3> Here is your order.";  
  var orderResults = document.gEBI("my-order")  
}
```

GREAT WORK!



We've linked the `var` to the `div`.

Now we need to say what we want to send.

Method: innerHTML

InnerHTML



Hi, my name is

Inner.

I am a HTML
Method.

MY SUPER-POWER:

I can send information to
anywhere on the webpage.

FOR EXAMPLE:

Send "results" with `var
orderResults` which gEBI linked
to `id:"my-order"` .

`orderResults.innerHTML = results;`

CHALLENGE

STEP 3

CODE:

Use **Inner** to show "results".
From `var orderResults` to the
"my-order" `<div>`.

```
function order() {  
  results = "<h3>Success!</h3> Here is your order.";  
  var orderResults = document.getElementById("my-order");  
  orderResults.innerHTML = results;  
}
```

EXPLAINER:

Inner is sending the "results" to where ever **var** is linked.

```
function order() {  
- results = "<h3>Success!</h3> Here is your order.";  
  var orderResults = document.getElementById("my-order");  
  orderResults.innerHTML = results;  
}
```

The diagram consists of a dashed cyan line that starts at the end of the 'results' variable assignment on the second line of code, moves right, then down, then left, and finally up to point at the 'innerHTML' property of the 'orderResults' variable on the third line. A small cyan arrow points upwards from the end of the dashed line to the 'innerHTML' property.

CHECK



1. Do you see the results section on your website?



2. When you click the "Process Order" button does it show "Success!" ?

CHECKPOINT

Your JS code should look like this:

```
<script>
  function placeOrder() {
    results = "<h3>Success</h3> Here your order.";
    var orderResults = document.getElementById( "my-order" );
    orderResults.innerHTML = results;
  }
</script>
```


CHECKPOINT

Your CSS code
should look like this:

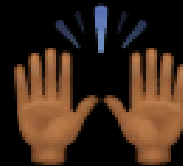
```
/* CSS ID */

#my-order {
  background-color: #29C1C4;
  padding: 25px;
}

/* HTML FORM BUTTON */


```

Here's a breakdown of the
program you are writing:



BREAK-DOWN

1.1.2

P3 Training 4

1.



2.



3.



4.



MY PROGRAM:

```
onclick="placeOrder();"
```

Event inside button.

```
function placeOrder() {  
}
```

Button click runs **function**.

```
var orderResults =
```

var **sends data** to <div>.

```
<div id="my-order">
```

<div> **shows** results.

CONGRATULATIONS



You've finished
Training 4.

QUIZ

1. 2.



ANSWER: NEXT SLIDE

QUESTION 3.4.1

Trigger a function with a click:

A. `<div class="button"> Contact </div>`

B. Insert in CSS: `onclick="ID"`

C. Link function to ID name in onclick event.

D. None of the above.

QUIZ

1. 2.



PREVIOUS ANSWER: C.

QUESTION 3.4.2

What's a variable like?

A. A dog chasing a cat.

B. A storage box with a special name.

C. A boat on the ocean.

D. No one really knows.

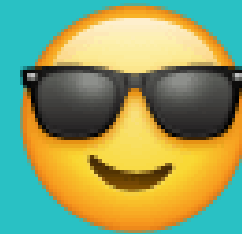
QUIZ

WHAT WAS YOUR SCORE?

1. 2.

50% 100%

PREVIOUS ANSWER: B.



It's time for Training 5
You will add some content to
your JS program.

Are you ready?



THE RULES:



DO:



- **Learn** to code.
- Start a **Coding League**.
- **Partner** with us.
- **Help** other people.
- Tell us how we can **improve**.
- **Use: #CodeJIKA** on all posts & publications.



DON'T:



- Say you've started a coding program and **not mention** #CodeJIKA.
- Modify our material or **say it's your own**.
- Issue press releases without mentioning #CodeJIKA .
- Be a **stinky-poo**.

LOADING . . .



PROJECT 3

Training 5

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

Desktop - v3 2019

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

Build an awesome website for a [Small Business](#) with a simple JavaScript order form.



LEARN:


The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 3 Hours	
3. Simple Biz Website // 6 Sessions 4.5 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

Diagram description: A box on the right side of the table, containing a certificate icon and the text '<Certificate> Milestone Project'. Two arrows point from this box to the 'Simple Biz Website' and 'Basic Business Website' rows of the table.

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info
2. Learn Flex, add “Cards” & content.
3. Build tables & add a form.
4. **JS I** Send & receive information from HTML.
5. **JS I** Add input from form & play hide & seek.
6. Footer, TEAM section & content writing.

BOM DIA*.



*Portuguese: Good morning.

TRAINING SNAPSHOT

PROJECT 3 - TRAINING 4

Missions for today.

Create an **automatic** order **summary**:

1. **Link variable** to <form>.
2. List **user responses** from <form>.
3. Play **hide and seek** with the results section.

MISSION

Link variable to `<form>`.



This mission will include 1 challenge.

Pull user responses
from the form
into the
`var orderForm` .

To **pull data** into a variable we need two things;

1. **HTML Method** to pull data.

2. Say from where (what **ID**).



```
var orderForm = document.getElementById("my-form");
```

CHALLENGE

STEP 1

CODE:

Create a new `var` called `orderForm` and using **gEBI**, link it to the "my-form" ID.

```
function order() {  
  var orderForm = document.getElementById("my-form");  
  results = "<h3>Success!</h3> Here is your order.";  
  var orderResults = document.getElementById("my-order");  
}
```

TIP:

Make sure the new `var` is:

A. On the first line of function.

B. Within `{ }` curly brackets.

NICE.

Now for the most exciting part:

Add the details from the
`<form>` to the `orderForm` var

MISSION

List **user responses** from `<form>`.



This mission will include 5 challenges.

CHALLENGE

STEP 1

WHERE:

Before `var orderResults` .

CODE:

Add a `
` and a caption "Name: " to the results using `results +=` .

```
var orderForm = document.getElementById("my-form");
results = "<h3>Success!</h3> Here is your order.";
results += "<br>Name: "
var orderResults = document.getElementById("my-order");
orderResults.innerHTML = results;
```

CHALLENGE

STEP 2

TIP:

Use + and always end with ;
semi-colon.

CODE:

Use ID `my-name` and `.value` to pull
the input.

```
results = "<h3>Success!</h3> Here is your order.";
results += "<br>Name: " + orderForm.elements["my-name"].value;
```


TIP:

- A. Use `+=` and always end with `;`.
- B. Also, note we're using `[]` brackets.

CHECK



1. Is the "Name" input from the form showing in the results?

(After you click "Process Order" button.)

TIP: Use the F5 key to refresh your page.

CHALLENGE

STEP 3

CODE:

Repeat with:

Caption: "**
Address: "**

ID: **my-address**

```
results = "<h3>Success!</h3> Here is your order.";
results += "<br>Name: " + orderForm.elements["my-name"].value;
results += "<br>Address: " + orderForm.elements["my-address"].value;
```

CHALLENGE

STEP 4

CODE:

Repeat with:

Caption: "**
I like to order: "**

ID: **my-drink**

```
results = "<h3>Success!</h3> Here is your order.";
results += "<br>Name: " + orderForm.elements["my-name"].value;
results += "<br>Address: " + orderForm.elements["my-address"].value;
results += "<br>I like to order: " + orderForm.elements["my-
drink"].value;
```

COME ON.

You can do it.

It's the last one.



CHALLENGE

STEP 5

CODE:

Repeat with:

Caption: "**
Quantity:** "

ID: **my-qty**

```
results = "<h3>Success!</h3> Here is your order.";
results += "<br>Name: " + orderForm.elements["my-name"].value;
results += "<br>Address: " + orderForm.elements["my-address"].value;
results += "<br>I like to order: " + orderForm.elements["my-
drink"].value;
results += "<br>Quantity: " + orderForm.elements["my-qty"].value;
```

MISSION

Play **hide and seek** with the results section.



This mission will include 7 challenges.

OK.

Now, we want to **hide the results** section until the order is submitted.

How will we do that?

1. Put a `display: none;` into the CSS ID.
(This **hides** the section.)
2. Override with HTML **style** once button is clicked.
(This **shows** the section.)

`orderResults.style.display = "block";`



CHALLENGE

STEP 1

CODE:

Hide the results by adding `display: none;` to the CSS `#my-order` ID.

```
#my-order {  
  background-color: #29C1C4;  
  padding: 25px;  
  display: none;  
}
```

Style.display

.style.display =



Hi, my name is

Styl.

I am a HTML
DOM.

MY SUPER-POWER:

I can make information **appear** or **disappear** anywhere on the webpage.

FOR EXAMPLE:

In `var orderResults` I can use "block" or "none" to make results show or hide.

`orderResults.style.display = "block";`

CHALLENGE

STEP 2

WHERE:

Before `innerHTML`.

CODE:

Use **style** to make the hidden results appear.

In `var orderResults`.

```
var orderResults = document.getElementById("my-order");  
orderResults.style.display = "block";  
orderResults.innerHTML = results;
```

CHECK



1. Is the results section hidden on the webpage?



2. When you click the "Process Order" button does show?

TIP: Use the F5 key to refresh your page.

DONE.



Was this hardest lesson so far?

CHECKPOINT

Your code should look like this:

```
<script>
  function placeOrder() {
    var orderForm = document.getElementById("my-form");
    results = "<h3>Success</h3> Here your order.";
    results += "<br>Name: " + orderForm.elements["my-name"].value;
    results += "<br>Address: " + orderForm.elements["my-address"].value;
    results += "<br>I like to order: " + orderForm.elements["my-
drink"].value;
    results += "<br>Quantity: " + orderForm.elements["my-qty"].value;
    var orderResults = document.getElementById("my-order");
    orderResults.style.display = "block";
    orderResults.innerHTML = results;
  }
</script>
```

CHECKPOINT

Your CSS code
should look like this:

BONUS: ----->

```
/* CSS ID */

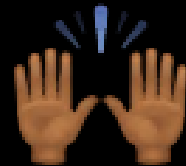
#my-order {
background-color: #29C1C4;
padding: 25px;
display: none;
}
```

```
/* HTML FORM BUTTON */
```

```
<input class="button" type="button" value="Process  
Order" onclick="placeOrder();" >
```

```
<input class="button" type="reset" value="Clear"  
onclick="document.getElementById('my-  
order').style.display = 'none';" >
```


Here's a breakdown of the
program you are writing:



BREAK-DOWN

1.1.2

P3 Training 4

1.



2.



3.



4.



5.



MY PROGRAM:

```
onclick="placeOrder();"
```



```
function placeOrder() {  
}
```

```
var orderForm =
```

```
var orderResults =
```

```
<div id="my-order">
```

Event inside button.

Button click runs **function**.

var **pulls user** info.

var **sends data** to <div>.

<div> **shows** results.

BONUS



JS APPRENTICE
200 POINTS

CODE:

- Add the line (button).
- Figure out what each part does.

TIP: Remember " at the end.

```
<input class="button" type="button" value="Process Order"  
onclick="placeOrder();" >
```

```
<input class="button" type="reset" value="Clear"  
onclick="document.getElementById('my-order').style.display = 'none';" >
```

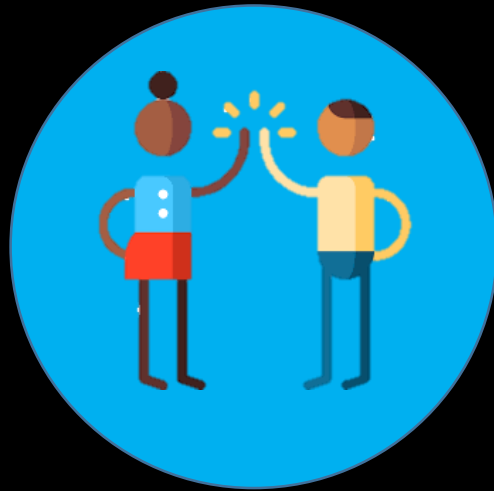
TIP:

Notice there are:

A. (') Single quotes.

B. (") Double quotes.

CONGRATULATIONS



You've finished
Training 5.

QUIZ

1. 2.



ANSWER: NEXT SLIDE

QUESTION 3.5.1

getElementById

- A. Collects information from CSS.
- B. Is an HTML form element to send user data.
- C. Get info from an element in your document.
- D. All of the above.

QUIZ

1. 2.



PREVIOUS ANSWER: C.

QUESTION 3.5.2

InnerHTML

- A. Links user data with JS functions.
- B. Allows a form to show anywhere.
- C. Sets/returns HTML content of an element.
- D. A compound JS term to confuse you.

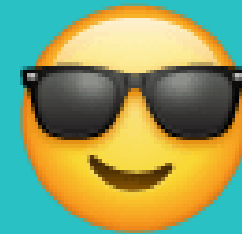
QUIZ

WHAT WAS YOUR SCORE?

1. 2.

50% 100%

PREVIOUS ANSWER: C.



Training 6 is really easy
and it's the last one.

Are you ready?



LOADING . . .



CODEJIKA.COM

PROJECT 3

Training 6

PRONOUNCED: [CODE-GEE-KA](#)

JIKA MEANS "DANCE" OR TURN IN ZULU, A LANGUAGE OF SOUTH AFRICA.

Desktop – v6

LEARN MORE: WWW.CODEJIKA.COM

WHAT'S THE DEAL?

PROJECT 3:

Build an awesome website for a [Small Business](#) with a simple JavaScript order form.



LEARN:


The [practical skills](#) of a Junior Web Developer over 12 Projects.

PURPOSE:

Teach yourself. Teach your friends. [Help others.](#)

PROJECTS OVERVIEW:

1. Simple Landing Page // 4 Sessions 90 Minutes	
2. CV Website // 6 Sessions 2.5 Hours	
3. Simple Biz Website // 5 Sessions 4 Hours	
4. JavaScript OrderForm // 4 Sessions 3 Hours	
5. Styling your Website // 4 Sessions 3 Hours	
6. Basic Business Website // 5 Sessions 4 Hours	



<Certificate>
Milestone
Project

Diagram description: A box on the right contains a certificate icon and the text '<Certificate> Milestone Project'. Two arrows point from this box to the 'Simple Biz Website' and 'Basic Business Website' rows in the table above.

PROJECT 3 – SIMPLE BIZ WEBSITE

TRAINING SESSIONS:

1. Setup: Header & Business Info
2. Learn Flex, add “Cards” & content.
3. Build tables & add a form.
4. **JS** | Send & receive information from HTML.
5. **JS** | Add input from form and play hide & seek.
6. Footer, TEAM section & content writing.

TRAINING SNAPSHOT

PROJECT 3 – TRAINING 6

These are your missions for today.

Create footer and add more content.

1. Create a stylish **footer**.
2. Write **content** for the business.
3. Add a “**Team**” section.

MISSION

Create a stylish **footer**.



This mission will include 8 challenges.

CHALLENGE

STEP 1

CODE:

Create a class called “footer” with 30px padding and a grey background.

TIP:

```
padding: ???;  
background: ???;
```


CHALLENGE

STEP 2

CODE:

Create a `<div>` and link it to `.footer` class styling.

WHERE:

Immediately above the `<script>` section.

CHALLENGE

STEP 3

CODE:

Write this in the `<div>` in an `h3` tag:

```
Thanks for visiting.
```

WHERE:

In the `<div>` you just created.

KEEP IT UP!

CHALLENGE

STEP 4

CODE:

1. Make another `<div>` inside the `<div>` you just created.
2. Apply the `.container` class styling to it.

WHERE:

Immediately below the `</h3>` closing.

CHALLENGE

STEP 5

CODE:

1. Create two `<article>`s in the same `<div>`.
2. Style both with the `“.card”` class.

TIP:

Make sure to close the first one before opening the next one.

THAT'S PROGRESS.



CHALLENGE

STEP 6

CODE:

Add this in the first `<article>`:

In `<h3>`: Get in Touch.

In `<p>`: Probably the best place to buy cupcakes in all of Joburg.

CHALLENGE

STEP 7

CODE:

Add this to the second `<article>`:

In H3: Contact Us:

In p:

We would love to hear from you.

`
` Call or email us. `
`

Mobile: 072345 633 //

Tel: 012 343 3432

CHALLENGE

STEP 8

CODE:

Add a copyright line with `©`, the year and company name.

WHERE:

The bottom of the footer. Right before the `</div>` closing tag of the footer.

FOOTER COMPLETE.

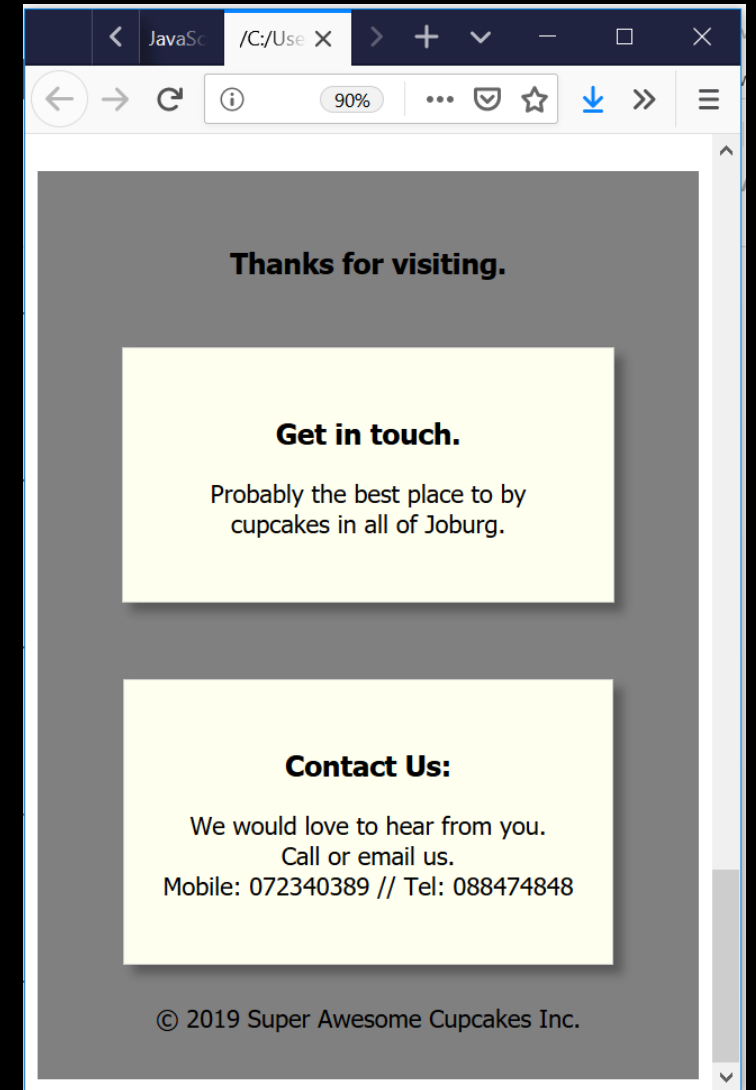
Hop on one foot to celebrate.



DESIGN OVERVIEW:

It should look something like this.

Feel free to swap the color and, of course, change the text.



MISSION

Define the services of the business.



This mission will include 1 **VERY IMPORTANT** bonus challenge.



BUSINESS COACH: WANNA BE A GREAT WEBDESIGNER?



Learn about your client ----->

Who are you making the website for?



Explain their business well.



Use engaging **Photos** and text.

TIP: Ask a friend to edit your text or shoot photos for you.



CONTENT CREATION:

Content is a boring name for

awesome stuff.



Content = Videos.

Photos.

GIFs.

Poems.

Explanations.

Any other type of text.




BUSINESS COACH: WEBSITE CONTENT CHECKLIST:



“I created a checklist to help you discover and write about what the best part of the business.”

NOTE: This is in your folder as a PDF or you can download from www.codejika.com .

PRINT THIS PAGE.

 CODEJIKA.COM

LEVEL 1: BUSINESS WEBSITE CONTENT LIST

1. WARM-UP:

What is the purpose of your business?
What is special about your business?
What is the most important product or service in your business?
Why did you start this business?

2. CORE:

How do you describe your business in one sentence?
Option 1: _____
Option 2: _____
When was the business started? _____
How many clients do you have? (Per month or in total.) _____

3. DETAILS:

Address: _____
Phone: _____
Email: _____
Preferred Web Domain (Example: www.DaveDonut.com or www.CharlieIndustries.com)
1: _____ 2: _____ 3: _____

Page 1 of 2

LEVEL 1: BUSINESS WEBSITE CONTENT LIST

1. WARM-UP:

What is the purpose of your business?

What is special about your business?

What is the most important product or service in your business?

Why did you start this business?

2. CORE:

How do you describe your business in one sentence?

Option 1. _____

Option 2. _____

When was the business started? _____

How many clients do you have? (Per month or in total.) _____

3. DETAILS:

Address: _____

Phone: _____

Email: _____

Preferred Web Domain (Example: www.DaveDonut.com or www.CharlieIndustries.com)

1.: _____ 2.: _____ 3.: _____

4. TEAM:

(Optional: Get a simple, friendly photo of each person's face and shoulders.)

Person 1:

Name: _____ Designation: _____

One sentence about what this person is good at and why:

Person 2:

Name: _____ Designation: _____

One sentence about what this person is good at and why:

1. CUSTOMER REVIEWS:

(Optional: Get a simple, friendly photo of each person's face and shoulders.)

Customer 1:

Name: _____ From where: _____

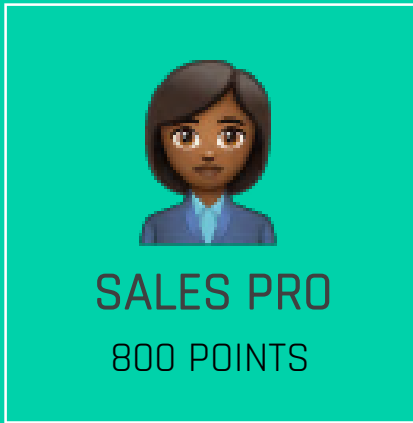
A few words about what this customer likes about business or their experience:

Customer 2:

Name: _____ From where: _____

A few words about what this customer likes about business or their experience:

BONUS CHALLENGE



BUSINESS:

- A. Design a website for a business
(or organization, club or school.)

- B. Use the checklist to
create the content.

A screenshot of a checklist form titled "LEVEL 1: BUSINESS WEBSITE CONTENT LIST" from CODEJKA.COM. The form is divided into three sections: 1. WARM-UP, 2. CORE, and 3. DETAILS. Each section contains several questions and blank lines for answers. The form is printed on a white background with a light blue header.

TIPS:

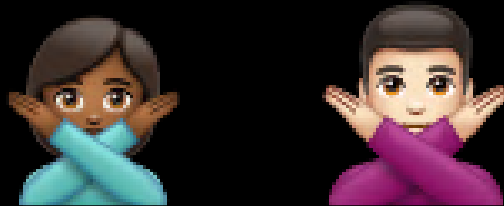
The checklist is attached at the end of the lesson.
Just print those two pages.

IMPORTANT:

NEVER, EVER meet a client alone.

Be careful and suspicious, especially of people that are overly friendly.

You can usually get all your information over the phone or email.



MISSION

Add a “Team” section.



This mission will include 4 challenges.

USE IT TWICE!

Create a new section of the website by copying and pasting a section of code.

Then, we'll change the content
(images and text) and the
background.

CHALLENGE

STEP 1

CODE:

Duplicate: Copy and paste the whole `<section>` with “OUR SERVICES” in.

From the opening to the closing tag.

WHERE:

Paste it immediately below the first section's closing tag `</section>`.

CHALLENGE

STEP 2

CODE:

Change the second section into a "TEAM" section.

Change "OUR SERVICES" sub-title to: "OUR TEAM".

CHALLENGE

STEP 3

*DEFINITION:

“Bio” says what a person is good at, studied or accomplished.

CODE:

Present two members of the team (in the cards.)

1. Change the emoji code.
2. `<h3>`: First and Last Name.
3. `<p>`: Add a short bio*.
4. `<button>`: Remove.

GOOD!

Now **change** the **background** color of the section by adding a bit of in-line **styling**.

CHALLENGE

STEP 4

CODE:

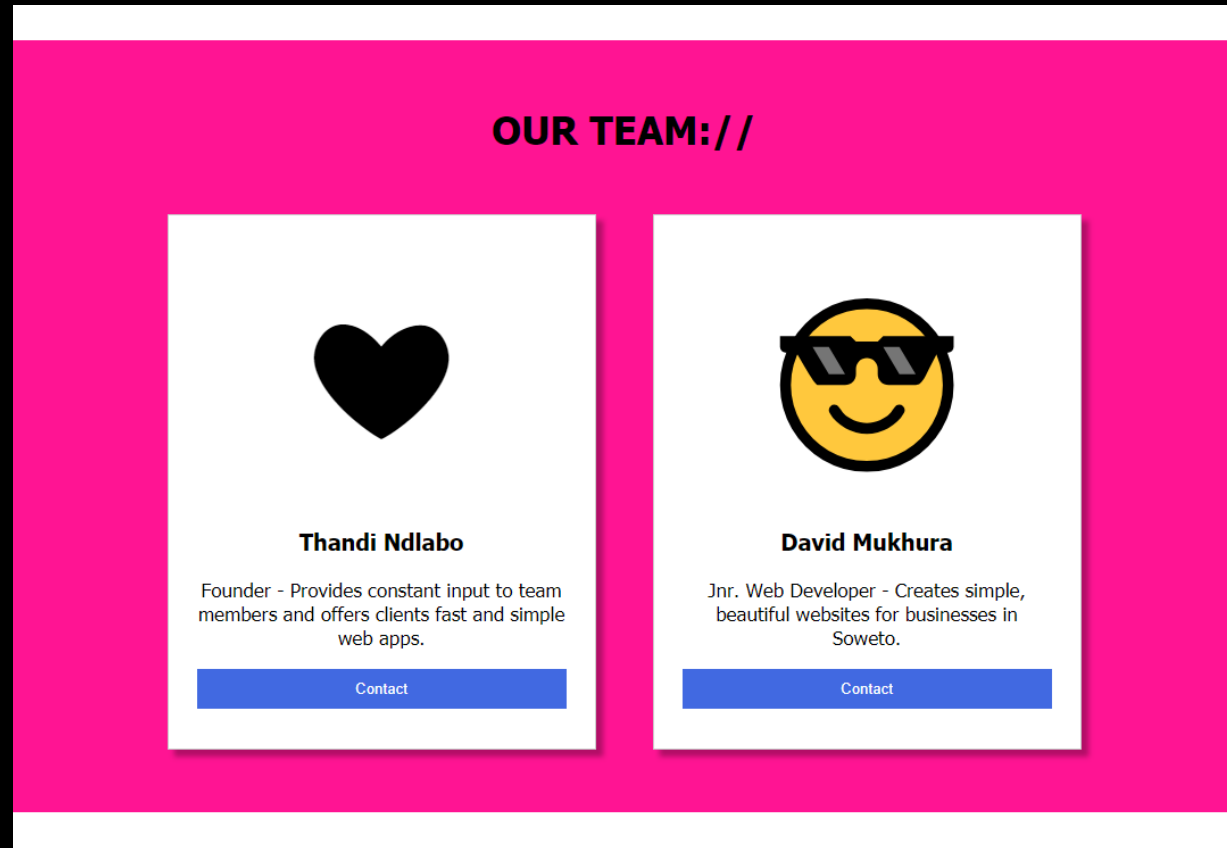
Add `style="background-color: deeppink;"` to the "TEAM" `<section>` opening tag.

```
<section class="container" style="background-color: deeppink;">
```

CHECK



The "Team" section can look something like this.



BONUS

CHALLENGE

STEP 1



CONTENT BOSS

45 POINTS

CONTENT:

1 of 2

1. Create a another new section for the website.
2. Decide what title and content should be there.

BONUS

CHALLENGE

STEP 2



CONTENT BOSS

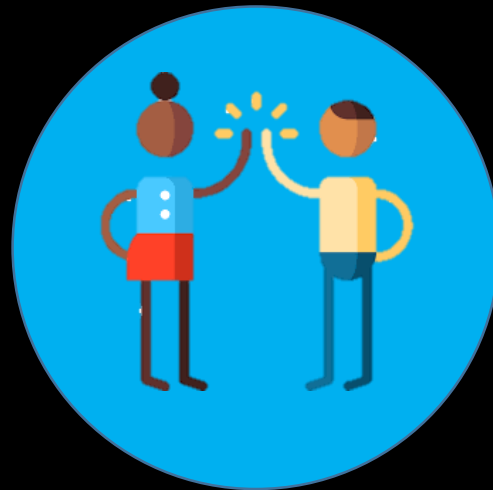
45 POINTS

CHECK:

2 of 2

- ✔ It should have 3 `<section>`s with cards and content.
- ✔ Make sure emojis (or photos) subtitle and paragraph are all unique in each section.

CONGRATULATIONS



You've finished
Training 5.

Here's another
simple 2-step bonus.

BONUS

CHALLENGE

STEP 1



SOCIAL LINKS

25 POINTS

CODE:

1 of 2

Insert this above the copyright line.

```
<div class="circle">F</div>  
<div class="circle">T</div>  
<div class="circle">I</div>
```


BONUS

CHALLENGE

STEP 2



SOCIAL LINKS

25 POINTS

CODE:

Add this class.

```
.circle {  
  color: white;  
  font-size: 2.5em;  
  display: inline-block;  
  height: 40px;  
  width: 40px;  
  margin: 15px;  
  border-radius: 50%;  
  background: deeppink;  
  padding: 5px 10px 15px 10px;  
  font-family: helvetica;  
}
```

2 of 2

You just finished **PROJECT 3.**



That's **AMAZING.**
You should celebrate!!!

HOW CAN WE DO BETTER?



Post what you would improve
on Social Media or email us.

Use the hashtag #CodeJIKA .

BYE.

See you soon.