PROJECT 3

Yipppeeeeee!
PROJECT 3
Training 1

PURPOSE:
Teach yourself. Teach your friends. Help others.

LEARN:
The practical skills of a Junior Web Developer over 12 Projects.

WHAT’S THE DEAL?

PROJECT 3:
Build an awesome website for a Small Business with a simple JavaScript order form.

LEARN MORE: www.CODEJIKA.com

Desktop – v1

PRONOUNCED: CODE-GEE-KA
JIKA MEANS "DANCE" OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

www.codejika.com

2019 Copyright Code for Change | Ultimate Growth
# PROJECTS OVERVIEW:

<table>
<thead>
<tr>
<th>Project</th>
<th>Sessions</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Simple Landing Page</td>
<td>4</td>
<td>90 Minutes</td>
</tr>
<tr>
<td>2. CV Website</td>
<td>6</td>
<td>2.5 Hours</td>
</tr>
<tr>
<td>3. Simple Biz Website</td>
<td>5</td>
<td>4 Hours</td>
</tr>
<tr>
<td>4. JavaScript OrderForm</td>
<td>4</td>
<td>3 Hours</td>
</tr>
<tr>
<td>5. Styling your Website</td>
<td>4</td>
<td>3 Hours</td>
</tr>
<tr>
<td>6. Basic Business Website</td>
<td>5</td>
<td>4 Hours</td>
</tr>
</tbody>
</table>
## PROJECT 3 – SIMPLE BIZ WEBSITE

### TRAINING SESSIONS:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Setup: Header &amp; Business Info</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Learn Flex, add “Cards” &amp; content.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Build tables &amp; add a form.</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td><strong>JS</strong> I Send &amp; receive information from HTML.</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td><strong>JS</strong> I Add input from form and play hide &amp; seek.</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Footer, TEAM section &amp; content writing.</td>
<td></td>
</tr>
</tbody>
</table>
TRAINING SNAPSHOT

These are your missions for today.

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

This mission will include 2 challenges.
START.

Start by setting up the skeleton of your code.
CHALLENGE

STEP 1

TIP:
Save a new notepad file as .html.

CODE:
Add this section:

```html
<!DOCTYPE html>
<head>
  <style>
  
  </style>
</head>
```
Now add `<header>` and `<section>` in `<body>`.

```html
<body>
 <header>
  </header>
 <section>
  </section>
</body>
```
GREAT JOB!
THINK

What's the name of your business?

Or who are you making it for?
Add a simple header, h1 & sub-title.

This mission will include 5 challenges.
Add an `<h1>` in `<header>` with your company name or brand.

```html
<h1>
    Awesome Industries Inc.
</h1>
```
CODE:

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

```html
<h4>Cute Cupcakes for Best Besties</h4>
```
NICE.

Keep it up.
Now let's style the `<body>` and `<header>`.
Add a body selector in CSS with these styles:

```css
body {
  font-family: tahoma;
  text-align: center;
}
```
For your Company name make it **white** and have the letters **spaced-out**.
Add `header { }` with a background color and some padding:

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

STEP 5

CODE:

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

TIP:

`color: ???;`
CHECKPOINT

Your code should look like this:

```html
<!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.
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.
Write a short Vision Statement.

Here is an example:

VISION: To create beautiful cupcakes for clients in the Randburg area.
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 `<br>` tags.

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

Create an “About Us” section.
Create an “About Us” explanation.

Here is an idea:

We've served over 100 clients and delivered over 10,000 tasty treats.
CODE:
Add the “About Us” content in a `<p>`.

WHERE:
After the VISION and the 2 `<br>`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.
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:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Awesome Industries Inc.</title>
  <style>
    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>
</body>
</html>
```

NOTE: Content should be your own.
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 . . .
PROJECT 3

Training 2

PRONOUNCED: CODE-GEE-KA
JIKA MEANS “DANCE” OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.

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

---

[Image] 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** 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

These are your missions for today:

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

This mission will include 3 challenges.
Here we go.
Flexbox

Flex allows you to easily space boxes or images.

Example: For an image gallery.
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**.

```html
<section class="container">
  <article></article>
  <article></article>
  <article></article>
</section>
```
STEP 2: Add the “Flexbox” Magic.

Container CSS

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

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

1. Container CSS

```css
.container {
  display: flex;
  justify-content: center;
}
```
### CSS FLEX REFERENCE:

<table>
<thead>
<tr>
<th>flex-direction</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;default&gt;</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flex-start</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>flex-end</td>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>center</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>space-between</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>space-around #1</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>space-around #2</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>stretch</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>space-evenly</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

REFERENCE: https://medium.com/@js_tut
LET’S FLEX!
CHALLENGE
STEP 1

TIP:
.container { } .

CODE:
Create a CSS class .container with:

```css
.container { 
  display: flex; 
  justify-content: center; 
  flex-wrap: wrap; 
}
```
THAT’S FLEX-TASTIC!
CHALLENGE

STEP 2

CODE:

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

WHERE:
Place below “Welcome” section closing tag</section>.

```html
<section class="container">
  <article> </article>
  <article> </article>
</section>
```
Insert a sub-title with `<h2>` at the top of the section: SERVICES://
Now we’re going to add a card design to the `<article>` s.
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.
BRIEFING

1.1.2

CARDS

Each “Card” has three elements inside:

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

2019 Copyright Code for Change | Ultimate Growth
CARDS

Your code will look like this:

```html
<article class="card">
  <div class="icons">имвол-символ</div>
  <h3>David Mukhura</h3>
  <p>Jnr. Web Developer - Creates beautiful websites for businesses in Soweto.</p>
  <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;
}
Add a padding and margin to the `.card` class.

```css
.card {
    border: 1px solid #ccc;
    background-color: ivory;
    margin: 25px;
    padding: 25px;
}
```
Adapt flex and the “Cards” to show nicely on larger screens.

Add this CSS:

```css
@media screen and (min-width: 50em) {
  .card {
    flex-basis: 325px;
  }
}
```
CHALLENGE
NOT RELATED TO “CARDS.”

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

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

TIP:
Blur means to make something fuzzy.
RGBA Transparency Trick

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

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

**TIP:**
Transparent means “See-through.”

**TIP 2:**
This doesn't work in all browsers.
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.

```css
.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.
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.
Create a CSS class called `.icons`.

```css
.icons {
  font-size: 8em;
  padding: 25px;
}
```
Create a `<div>` nested inside each `<article>`. Insert an emoji in your `<div>`.

```html
<article class="card">
  <div class="icons"> &#10084;
  </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:**

```html
<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:**

```html
<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:**

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

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

*Note: We’ll use the class later.*
Add 3 rules to the button CSS:

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

You're done.
/* 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;
}
<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:

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

```html
<section class="blue container">
</section>
```
BONUS

CHALLENGE

STEP 2

CLASSY BLUE

25 POINTS

CODE:

Add this class.

```
.blue {
  background: teal;
}
```
WELL DONE!

This was a long, hard lesson.
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.

ANSWER: NEXT SLIDE
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?

😊
Help other kids learn to code.

Post on Social Media what you’re working on with #CodeJIKA.
LOADING . . .
PROJECT 3
Training 3

PRONOUNCED: CODE-GEE-KA
JIKA MEANS “DANCE” OR TURN IN ZULU, A LANGUAGE OF SOUTH AFRICA.

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:**

<table>
<thead>
<tr>
<th>#</th>
<th>Project</th>
<th>Sessions</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Simple Landing Page</td>
<td>4</td>
<td>90 Min</td>
</tr>
<tr>
<td>2.</td>
<td>CV Website</td>
<td>6</td>
<td>2.5</td>
</tr>
<tr>
<td>3.</td>
<td>Simple Biz Website</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>JavaScript OrderForm</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Styling your Website</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Basic Business Website</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</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.
These are your missions for today:

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

This mission will include 4 challenges.
YALLA*.

*"Come on" in Arabic.
Tables consist of:

A. **Columns** `<th>`

B. **Rows** `<tr>`

<table>
<thead>
<tr>
<th>Column 1</th>
<th><strong>COLUMN</strong></th>
<th>Column 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROW</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: HTML Table
Tables are a great way to store and show information.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Themba</td>
<td>14</td>
<td>074 7858588</td>
</tr>
<tr>
<td>John</td>
<td>18</td>
<td>098 5211611</td>
</tr>
<tr>
<td>Rachael</td>
<td>17</td>
<td>062 3241279</td>
</tr>
</tbody>
</table>

Example: HTML Table
### Example: HTML Table

```
<table>
  <tr>
    <td>Name:</td>
    <td>Themba</td>
    <td>074 7858588</td>
  </tr>

  <tr>
    <td>Age:</td>
    <td>14</td>
    <td></td>
  </tr>

  <!-- Additional rows can be added here... -->
</table>
```
Options: Table Styling:

```html
<table class="t01">
  .t01 {
    width: 100%;
    background-color: #f1f1c1;
  }
</table>

<th>Table Header cell</th>
<td rowspan="2">Merge two cells</td>
</table>
```
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.

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

STEP 2

WHERE:
Below </h2> and before </section>.

CODE:
Build this table:

```html
<table>
  <tr>
    <td> Name: </td>
    <td> </td>
  </tr>
  <tr>
    <td> Address: </td>
    <td> </td>
  </tr>
</table>
```
### Challenge

**Step 3**

### Code:

Add two more rows:

```html
...<tr>
  <td> Favourite drink: </td>
  <td> </td>
</tr>
...<tr>
  <td> Quantity: </td>
  <td> </td>
</tr>
...```

**Where:** After row 2 and before the `</table>` closing.
BOOM!

You just built your first table.
WHERE: You should know where by now. ;)

CHALLENGE

STEP 4

CODE:

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

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

table {
    margin: auto;
}
```
Add a form to a table.

This mission will include 6 challenges.
Form is used to collect user* input*.

GOOFY FORM:
What’s the silliest idea you have ever had?

Example: HTML Form

Definitions:
*User*: Anyone who uses your website.
*Input*: Information given.
Example: Simple HTML Form

```html
<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 Actions

Example: Form Actions

```html
<form action="">
  <form id="my-form" action="me.html" target="_blank">
    <input type="text" placeholder="Enter text">
    <button type="submit">Submit</button>
  </form>
</form>
```

CSS ID styling #my-form

Where data is sent

_blank = new tab

_self = same page
Wrap the whole table in `<form>` tags.

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

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

Form Inputs

Example: Form Inputs

```html
<input type="text" size="50" name="idea" value="Idea...">
```

**Types:** Text, radio & submit.

**Width** input field in px.

**Name** for saving information.

**Placeholder** text.
### HTML INPUT TYPES

#### Set 1

<table>
<thead>
<tr>
<th>TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;input type=&quot;button&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;checkbox&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;color&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;date&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;datetime-local&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;email&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;file&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;hidden&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;image&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;month&quot;&gt;</code></td>
</tr>
</tbody>
</table>

#### Set 2

<table>
<thead>
<tr>
<th>TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;input type=&quot;password&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;radio&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;range&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;reset&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;search&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;submit&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;tel&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;text&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;time&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;url&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;week&quot;&gt;</code></td>
</tr>
</tbody>
</table>

More online here: [https://www.w3schools.com/html/html_form_input_types.asp](https://www.w3schools.com/html/html_form_input_types.asp)
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.

```html
<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.

```html
<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**

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

---

**DEFINITION:**

*Default*: The standard. This value remains in effect unless changes are made.
In ROW 4, add a “number” type of `<input>`.

Define a minimum of 1 and a maximum of 5.

```html
<tr>
  <td>Quantity:</td>
  <td>
    <input type="number" name="my-qty" value="1" min="1" max="5">
  </td>
</tr>
```
Add a short note below the `<input>` to notify the user of the limits.

Make it `<small>` with HTML.

```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.
Add some buttons to the form.

This mission will include 6 challenges.
Add two rows.

- We need a bit of space, so add a `<br>` in the fifth ROW (below the “Quantity:” ROW.)

```html
</tr>
<tr>
  <br>
</tr>
</table>
```
### CHALLENGE

**STEP 2**

### CODE:

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

```html
</tr>
<tr>
  <br>
</tr>
<tr>
  <td colspan="2">
  </td>
</tr>
</table>
```

<table>
<thead>
<tr>
<th>ROW 5</th>
<th>ROW 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Add a button in ROW 6 using `type="button"` and a value of “Process Order”.

```html
<tr>
  <td colspan="2" >
    <input
        type="button"
        value="Process Order" >
  </td>
</tr>
```
Add a second button with type="reset" and a value of “Clear”.

```html
<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:

```css
button, .button {...}
```

You’ll link to that styling in the next Challenge.
Add styling to BOTH the buttons using `class="button"`.

**WHERE:**
Within the `<input>` tags themselves.

**CODE:**

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

To make sure your buttons are reflecting the styling correctly.
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.
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.

ANSWER: NEXT SLIDE
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.

PREVIOUS ANSWER: C.
QUIZ

WHAT WAS YOUR SCORE?

1.  

2.  

☐ ☐

50% 100%

PREVIOUS ANSWER: A.

www.codejika.com
It’s time for Training 4 and some JavaScript.

Are you ready?
LOADING . . .
PROJECT 3
Training 4

PROJECT 3:
Build an awesome website for a Small Business with a simple JavaScript order form.

PURPOSE:
Teach yourself. Teach your friends. Help others.

LEARN:
The practical skills of a Junior Web Developer over 12 Projects.

WHAT'S THE DEAL?

LEARN MORE:
www.CODEJIKA.com

PRONOUNCED: CODE-GEE-KA
JIKA MEANS “DANCE” OR TURN IN ZULU, A SOUTH AFRICAN LANGUAGE.
Post on Social Media what you're working on with #CodeJIKA.

Help other kids learn to code.
### PROJECTS OVERVIEW:

<table>
<thead>
<tr>
<th>No.</th>
<th>Project Description</th>
<th>Sessions</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Simple Landing Page</td>
<td>4</td>
<td>90</td>
</tr>
<tr>
<td>2.</td>
<td>CV Website</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>Simple Biz Website</td>
<td>6</td>
<td>4.5</td>
</tr>
<tr>
<td>4.</td>
<td>JavaScript OrderForm</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Styling your Website</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>Basic Business Website</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
## PROJECT 3 - SIMPLE BIZ WEBSITE

### TRAINING SESSIONS:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup: Header &amp; Business Info</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Learn Flex, add “Cards” &amp; content.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Build tables &amp; add a form.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>JS</strong></td>
<td>Send &amp; receive information from HTML.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>JS</strong></td>
<td>Add input from form and play hide &amp; seek.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Footer, TEAM section &amp; content writing.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

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!!!
JavaScript | <script>

JavaScript is written between <script> tags.

<script>
    JavaScript Code
</script>

WHERE:
At the end of site before </body> closing tag.
Add script tags before the closing `</body>` tag.

```html
<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

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onclick</td>
<td>Fires on a mouse click on the element</td>
</tr>
<tr>
<td>onkeypress</td>
<td>Fires when a user presses a key</td>
</tr>
<tr>
<td>onmousemove</td>
<td>Fires when the mouse pointer is moving while it is over an element</td>
</tr>
<tr>
<td>onload</td>
<td>Fires after the page is finished loading</td>
</tr>
<tr>
<td>onoffline</td>
<td>Script to be run when the browser starts to work offline</td>
</tr>
<tr>
<td>onresize</td>
<td>Fires when the browser window is resized</td>
</tr>
<tr>
<td>oninput</td>
<td>Script to be run when an element gets user input</td>
</tr>
</tbody>
</table>

More online here: [https://www.w3schools.com/tags/ref_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp)

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

1.1.2

HTML Event: `onclick`

`onclick=""`

When *mouse* is *clicked*.

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

1.1.2

HTML Event: onclick

EXAMPLE:

A HTML Event will trigger a JS Function

...when button is clicked.
BRIEFING

HTML Event: onclick

1. HTML Element with onclick event

2. Button with onclick event

3. Function call with onclick event

3-STEP EXAMPLE:

1. onclick event in button.

2. Triggers function placeOrder().

3. Function placeOrder() is run.
Add the `onclick` HTML Event to the “Process Order” `<input>` with `placeOrder()` function.

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

Let's learn function.
IT WORKS LIKE THIS:

function placeOrder () {

    Name

    Calculations go in here.

It makes things happen.
REMEMBER

What's a function?
Add a function called "placeOrder ()".

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

JS: `placeOrder` (With a CAPITAL “O”.)
function placeOrder () {
  results = "<h3>Success!</h3> Here is your order.";
}

CODE:

Add a success message inside the function using:

```javascript
results = "<h3>Success</h3>..."
```
TIP:

JS:

Put a `;` semi-colon at the end of each line.
GOOD JOB.

You just finished mission 1.
Create the order summary section.

This mission will include 2 challenges.
MISSON BRIEFING

LINK THE MASTERS

CSS

An ID

HTML

to a <div>

JS

& a var.

#my-order
BRIEFING
JS | Variable

STRUCTURE:

Function uses Variable 2

...to send the results to div.

CSS ID

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

**CSS**

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

**HTML**

```
<div id="my-order">
</div>
```
Create a CSS ID called “my-order” with this styling:

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

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

STEP 2

WHERE:
Immediately before the </form> closing tag.

CODE:

Create a <div> with an “my-order” id inside.

```html
</tr>
</table>
<br>
<div id="my-order">
</div>
</form>
```
ALRIGHTY THEN.

What's a variable?
Send results: Variable & HTML Methods.

This mission will include 3 challenges.
A vehicle to **store data** and give commands.

**STORE DATA:**

“Yaay. I love learning.”

**GIVE COMMANDS:**

“Get me some food.”
IT WORKS LIKE THIS:

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

```javascript
var store = "I like cats."
```
JS | var

Variables can store data from tons of places.

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

**Variable**

**Data to store**

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

Add a JS variable called orderResults in the function.
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**).

```javascript
var orderResults = document.getElementById("my-order");
```
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.
BRIEFING
HTML Methods

HEROS WITH SUPER-POWERS.

Variable 2

A. Link to the ID name.

B. Show the results in HTML.
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?
Function `placeOrder()` {
    results = "<h3>Success!</h3> Here is your order.";
    var orderResults = document.getElementById("my-order");
}
GREAT WORK!

We’ve linked the var to the div.

Now we need to say what we want to send.
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;
function order() {
  results = "<h3>Success!</h3> Here is your order.";
  var orderResults = document.getElementById("my-order");
  orderResults.innerHTML = results;
}
Inner is sending the “results” to where ever var is linked.

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

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

2. When you click the “Process Order” button does it show “Success!”?
Your JS code should look like this:

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

"CHECKPOINT"
/* CSS ID */

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

/* HTML FORM BUTTON */

<input class="button"
    type="button"
    value="Process Order"
    onclick="placeOrder();" >
Here's a breakdown of the program you are writing:
Event inside button.

Button click runs function.

var sends data to <div>.

<div> shows results.
CONGRATULATIONS

You’ve finished Training 4.
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.
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.

PREVIOUS ANSWER: C.
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.
PROJECT 3

Training 5

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:

<table>
<thead>
<tr>
<th>Project</th>
<th>Sessions</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Simple Landing Page</td>
<td>4</td>
<td>90</td>
</tr>
<tr>
<td>2. CV Website</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3. Simple Biz Website</td>
<td>6</td>
<td>4.5</td>
</tr>
<tr>
<td>4. JavaScript OrderForm</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5. Styling your Website</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6. Basic Business Website</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</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 & play hide & seek.

6. Footer, TEAM section & content writing.
BOM DIA*.

*Portuguese: Good morning.
TRAINING SNAPSHOT

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.
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**).

```javascript
var orderForm = document.getElementById("my-form");
```
Create a new var called `orderForm` and using `gEBI`, link it to the “my-form” ID.

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

This mission will include 5 challenges.
Add a `<br>` and a caption “Name: “ to the results using `results += `.

```javascript
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.

```javascript
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: "<br>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: "<br>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: "<br>Quantity: "
ID: my-qty

```javascript
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;
```
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.)

   ```javascript
   orderResults.style.display = "block";
   ```
CHALLENGE
STEP 1

CODE:

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

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

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

STEP 2

WHERE:
Before `innerHTML`.

CODE:

Use `styl` to make the hidden results appear.

In `var orderResults`.

```javascript
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?
Your code should look like this:

```html
<script>
    function placeOrder() {
        var orderForm = document.getElementById("my-form");
        var orderResults = document.getElementById("my-order");
        var 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;
        orderResults.style.display = "block";
        orderResults.innerHTML = results;
    }
</script>
```
Your CSS code should look like this:

```css
/* 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'" />
```

BONUS:
Here's a breakdown of the program you are writing:
1. Event inside button.

2. Button click runs function.

3. var pulls user info.

4. var sends data to <div>.

5. <div> shows results.
A. Add the line (button).
B. Figure out what each part does.

TIP: Remember " at the end.

```html
<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.
getElementsBy**Id**

**QUESTION 3.5.1**

1. 2.

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.

**ANSWER: NEXT SLIDE**
 QUESTION 3.5.2

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.

PREVIOUS ANSWER: C.
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?
PROJECT 3
Training 6

PRONOUNCED: CODE-GEE-KA
JIKA MEANS “DANCE” OR TURN IN ZULU, A LANGUAGE OF SOUTH AFRICA.

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:

<table>
<thead>
<tr>
<th>Project</th>
<th>Sessions</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Simple Landing Page</td>
<td>4</td>
<td>90 Minutes</td>
</tr>
<tr>
<td>2. CV Website</td>
<td>6</td>
<td>2.5 Hours</td>
</tr>
<tr>
<td>3. Simple Biz Website</td>
<td>5</td>
<td>4 Hours</td>
</tr>
<tr>
<td>4. JavaScript OrderForm</td>
<td>4</td>
<td>3 Hours</td>
</tr>
<tr>
<td>5. Styling your Website</td>
<td>4</td>
<td>3 Hours</td>
</tr>
<tr>
<td>6. Basic Business Website</td>
<td>5</td>
<td>4 Hours</td>
</tr>
</tbody>
</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.
TRAINING SNAPSHOT
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.
Create a stylish footer.

This mission will include 8 challenges.
Create a class called “footer” with 30px padding and a grey background.

TIP:

```css
padding: ???;
background: ???;
```
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.
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.
Get in Touch.

Probably the best place to buy cupcakes in all of Joburg.
In H3: Contact Us:

In p:

We would love to hear from you.
<br>Call or email us. <br>
Mobile: 072345 633 //
Tel: 012 343 3432
**CODE:**
Add a copyright line with `&copy;`, 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.
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**.

“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.
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: ____________________________________________
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:
____________________________________________________
BUSINESS:

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

B. Use the checklist to create the content.

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.
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”.

2019 Copyright Code for Change | Ultimate Growth
**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.

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

---
GOOD!

Now `change` the `background` color of the section by adding a bit of in-line `styling`.
<section class="container" style="background-color: deeppink;">
CHECK

The “Team” section can look something like this.

OURS TEAM://

Thandi Ndlabo
Founder - Provides constant input to team members and offers clients fast and simple web apps.

David Mukhura
3rd. Web Developer - Creates simple, beautiful websites for businesses in Soweto.
1. Create a new section for the website.
2. Decide what title and content should be there.
CHECK:

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

You've finished Training 5.
Here’s another simple 2-step bonus.
CODE:

Insert this above the copyright line.

```html
<div class="circle">F</div>
<div class="circle">T</div>
<div class="circle">I</div>
```
Add this class.

```css
.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;
}
```
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.