(/)

Flexbox

CSS   **Responsive design**

⬆ Amateur
👤 By: David Dias, Senior Software Engineer at HomeX
⚙ Weight: 1
☑ Your score will be updated once you launch the project review.

(/)

Resources

Read or watch:

- A Complete Guide to Flexbox | CSS-Tricks (/rltoken/6jU5TDzpFyATZFezlGeYHg)
- Flexbox Froggy - A game for learning CSS flexbox (/rltoken/yAg48ol9NMFkHuusLWXa1Q)
- Flexbox Defense (/rltoken/KxhVsIGHDrl1NJlq3U5MPg)
- Flexbox Cheatsheet (/rltoken/o3PJNSz-zpny2tFVsM26cg)
- CSS Flexible Box Layout - CSS: Cascading Style Sheets | MDN (/rltoken/1fF-czARxONNFpmijXS_hw)
- afonsopacifer/awesome-flexbox: A curated list of CSS Flexible Box Layout Module or only Flexbox. (/rltoken/JxO15F7Sa6uxTvXRwVDU9A)
- Build with Flexbox (/rltoken/bA5Gc3lwygz2ep9-Ag380w)
- Flexplorer (/rltoken/Vjz54S7kjlgzjj26o0C2wQ)
- CSS Flexible Box Layout Module Level 1 (/rltoken/eEeqnOZcZNhW8wOgyq42EA)
- FLEX: A simple visual cheatsheet for flexbox (/rltoken/-oZvlNfsZ6gU47PvDEYC6Q)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/76owYORNCZaAjbCCSrFPcA), **without the help of Google**:

- What is Flexbox?
- How to convert float positioning to a flex display
- How to horizontally and vertically align elements using Flexbox
- The difference between the main and cross axes
- Properties that work on flex elements vs flex container
- Shorthands for flex
- How to create a new page with flex in mind

Files

Required images for your HTML files

Download the images linked in the **CSS Advanced** and place them into an images directory at the root of the project.

HTML starter file

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover">
  <title>Homepage - Techium</title>
  <meta name="description" content="Description of the page less than 150 characters">
  <link rel="icon" type="image/png" href="images/favicon.jpg">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700|Raleway:700&display=swap" rel="stylesheet">
  <link rel='stylesheet' href='styles.css'>
 </head>
 <body>
  <!– Header –>
  <header class="header" data-section-theme="dark">
   <div class="container">
    <div class="header-logo">
     <a href="#">
      <img src="images/logo-white.png" alt="Techium logo" width="160" height="40">
     </a>
```

CSS starter file

Click to expand/hide file contents

Files for tasks 10 and onward

(/)

article.html

```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, viewport-fit=cover">
  <title>Article - Techium</title>
  <meta name="description" content="Description of the page less than 150 characters">
  <link rel="icon" type="image/jpg" href="../images/favicon.jpg">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:400,700|Raleway:700&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="00-styles.css">
 </head>
 <body>
  <!-- Header -->
  <header class="header" data-section-theme="dark">
   <div class="container">
    <div class="header-container">
     <div class="header-logo">
      <a href="/">
       <img src="images/logo-white.png" alt="Techium logo" width="160" height="40">
```
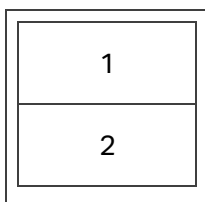
Quiz questions

Great! You've completed the quiz successfully! Keep going! (Hide quiz)

Question #0

What's the "flex" equivalent of:

```
┌─────────────┐
│      1      │
├─────────────┤
│      2      │
└─────────────┘
```

○ flex-direction: column

○ flex-direction: row

Question #1

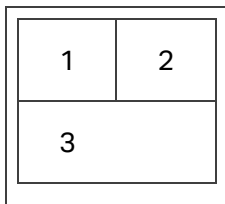Which of these CSS attributes are part of the "Flex container"?
 (/)
Select all correct answers

- ☐ order
- ☑ align-items
- ☐ flex
- ☑ flex-wrap
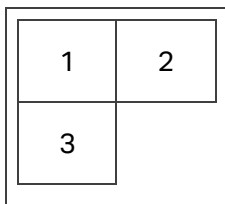- ☑ justify-content
- ☐ align-self

Question #2

What's the "flex" equivalent of:

```
┌─────────────────┐
│ ┌──────┬──────┐ │
│ │  1   │  2   │ │
│ ├──────┴──────┤ │
│ │  3          │ │
│ └─────────────┘ │
└─────────────────┘
```

- ○ Container: `flex-direction: row + flex-wrap: wrap` / last item: `flex-grow: 0`
- ○ Container: `flex-direction: row` / last item: `flex-grow: 1`
- ◉ Container: `flex-direction: row + flex-wrap: wrap` / last item: `flex-grow: 1`
- ○ Container: `flex-direction: column + flex-wrap: wrap`
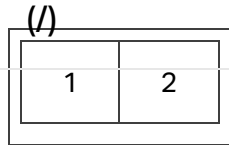
Question #3

What's the "flex" equivalent of:

```
┌──────────────────┐
│ ┌──────┬──────┐  │
│ │  1   │  2   │  │
│ ├──────┼──────┘  │
│ │  3   │         │
│ └──────┘         │
└──────────────────┘
```

- ○ Container: `flex-direction: column + justify-content: center`
- ◉ Container: `flex-direction: row + flex-wrap: wrap`
- ○ Container: `flex-direction: row` / last item: `flex-grow: 2`
- ○ Container: `flex-direction: row + flex-wrap: nowrap`

Question #4

What's the "flex" equivalent of:

(/)

1	2

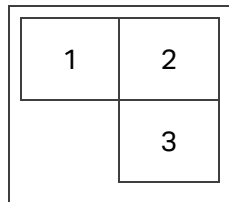○ flex-direction: column

◉ flex-direction: row

Question #5

Which of these CSS attributes are part of the "Flex items"?

Select all correct answers

☐ align-content

☐ flex-direction

☑ align-self

☑ flex-shrink

☑ order

Question #6

What's the "flex" equivalent of:

1	2
	3

○ Container: `flex-direction: row + flex-wrap: nowrap + justify-content: flex-end`

○ Container: `flex-direction: row + flex-wrap: wrap + justify-content: flex-start`

○ Container: `flex-direction: column + flex-wrap: wrap + justify-content: flex-end`

◉ Container: `flex-direction: row + flex-wrap: wrap + justify-content: flex-end`

Tasks

0. Add display flex   mandatory

Use the starter HTML and CSS files from this task to task 10. Copy the contents of the starter files into the files that you need to produce and make the necessary changes according to the task description.

When using `display: flex;` on a container, all direct children become `flex-items` (and no more inline or block elements).

With display flex, margins are not collapsing as they would with block items. Also remember that flexbox is 1-dimensional system (vs CSS Grid which is a 2 dimensional system)

In the `/* Grid` section within your CSS

- Add a selector for the `row` class
  - Property: `display` , Value: `flex`

=> Now, all children from the `row` class are flex items

- Entirely remove the `row::after` declaration
- Remove the `float: left` inside `[class*='col-']`

=> All elements should appear same than before using the float

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `0-index.html, 0-styles.css`

| Help | Check your code | >_ Get a sandbox | | **0/4** pts |

1. Add new classes on sections   mandatory

Using the files from the previous task as the base for this task:

In the outermost section tag for `services`

- Add the class `section-services`

In the outermost section tag for `works`

- Add the class `section-works`

In the outermost section tag for `about`

- Add the class `section-about-us`

In the outermost section tag for `latest_news`

- Add the class `section-latest-news`

In the outermost section tag for `testimonial`

- Add the class `section-testimonial`

In the outermost section tag for `contact`

- Add the class `section-contact`

Repo:

- GitHub repository: `alx_html_css`
(/) - Directory: `flexbox`
  - File: `1-index.html`, `1-styles.css`

| Help | Check your code | >_ Get a sandbox | **0/6** pts |

2. Reverse order Latest news cards   `mandatory`

Using the files from the previous task for this task:

The `flex-direction` property says how flex items are placed on the main axis and their direction (normal or reversed).

`flex-direction` is sometimes used when doing responsive design. Some elements may appear better when they are in column mode on mobile and row when on desktop.

`row-reverse` and `column-reverse` should be used in specific situation. The visual order of elements should be the same visually and in the HTML code. Refer to flex-direction - CSS: Cascading Style Sheets | MDN (/rltoken/IwIiIY5UOdA2LDwqRIxFrQ) for more information

In your CSS file:

Before `/*** 4. CARD ***/`, add a new comment: `/* Section Latest news ============================= */`

Under that comment section, target the `row` class inside `section-latest-news` class

- Property: `flex-direction`, Value: `row-reverse`

The Latest news should appear in a reverse order.

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `2-index.html`, `2-styles.css`

| Help | Check your code | >_ Get a sandbox | **0/3** pts |

3. Simplify services list   `mandatory`

Using the files from the previous task for this task:

`flex-wrap` is a property that can force the flex items to be in one or multiple lines. Learn more about it here (/rltoken/K9Wu7ow8fAYSJeX27yWwDQ).

In the `Services section` of `3-index.html`, remove the second `ul` and put the 3 `li` elements under the first `ul`

Now, in your CSS file, before `/*** 4. CARD ***/`, add a new comment: `/* Section SERVICES ============================ */`

Under that comment section, add a new selector targeting the `row` class inside the `section-services` class

- Property: `flex-wrap`, Value: `wrap`

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `3-index.html`, `3-styles.css`

| Help | Check your code | >_ Get a sandbox | **0/3** pts |

4. Playing around with the spacing between flex service items   `mandatory`

Using the files from the previous task for this task:

In `4-styles.css` file, within the Grid section

- In `.col-1-3` selector

  - Replace the current width with `calc((100% / 3) - 2rem)`
- In `.col-1-2` selector

  - Replace the current width with `calc((100% / 2) - 2rem)`
- In `[class*='col-']`

  - Remove the padding declaration
  - Set Property: `margin` to `1rem`
- In `ul.row` declaration

  - Replace the current margin with `-1rem`

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `4-index.html`, `4-styles.css`

| Help | Check your code | >_ Get a sandbox | **0/5** pts |

5. Flexify the header   `mandatory`

Using the files from the previous task for this task:

In your `5-index.html` file, inside the Header section (/)

- Wrap the div with class `header-logo` and the div with class `navbar-menu` with a `div` that has `header-container` as a class

In your `5-styles.css` file,

- Inside the `/* Header` section
- Add a selector for the `header-container` class
  - Property: `display`, Value: `flex`
  - Property: `justify-content`, Value: `space-between`
- Remove `header-logo` and `header-logo a` rules
- Remove the `navbar-menu` rule

- In the variables section

  - Remove
    - `header-logo-position`
    - `header-logo-link-display`
    - `header-logo-link-position`
    - `header-logo-link-top`
    - `header-logo-link-left`

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `5-index.html`, `5-styles.css`

Help | Check your code | >_ Get a sandbox   **0/11** pts

6. Flexify the navbar   `mandatory`

Using the files from the previous task for this task:

in `6-styles.css`, inside the `/* Navbar` section

- In the `nav` class selector
  - Property: `display`, Value: `flex`
- Inside the `.nav .nav-item` selector, remove the display declaration
- Target `.nav-item + .nav-item` inside `nav` class
  - Move the margin declaration from `.nav .nav-item` inside the new selector.
- In the variables section
  - Change the value of the variable `nav-item-margin` to be `0 0 0 2rem`

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`

- File: `6-index.html, 6-styles.css`

(/)

| Help | Check your code | >_ Get a sandbox | | **0/6** pts |

7. Align center logo and navbar   mandatory

Using the files from the previous task for this task:
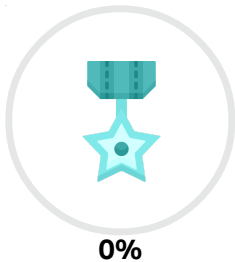
In `7-styles.css`, inside the `/* Header` section, in the `header-container` class selector, set the property `align-items` to `center`

Repo:

- GitHub repository: `alx_html_css`
- Directory: `flexbox`
- File: `7-index.html, 7-styles.css`

| Help | Check your code | >_ Get a sandbox | | **0/1** pt |

Score

0%

Your score will be updated once you launch the project review.

Please review **all the tasks** before you start the peer review.

◄ Review all the tasks   ►► Skip this project

Previous project (/projects/2048)