# CSS Layout - Flexbox – Lab

## Topics Covered

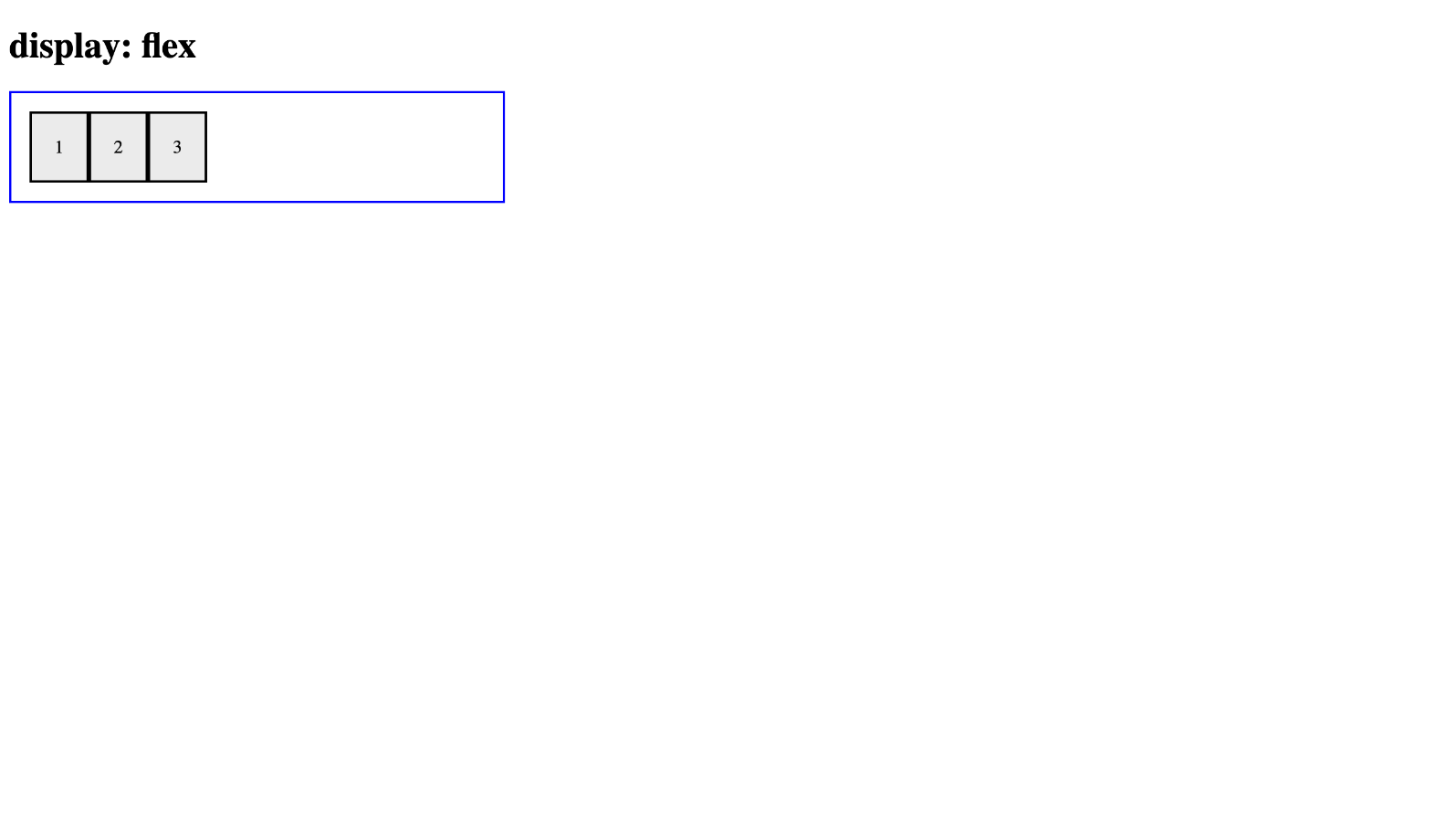
[Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout) is a one-dimensional layout method for arranging items in rows or columns. Items *flex* (expand) to fill additional space or shrink to fit into smaller spaces. This article explains all the fundamentals.

## Resources

* **Flexbox MDN**
  + <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox>
* **A Complete Guide to Flexbox**
  + <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
* **An Interactive Guide to Flexbox**
  + <https://www.joshwcomeau.com/css/interactive-guide-to-flexbox/>
* **Flexbox cheat sheet**
  + <https://codepen.io/kdankov/pen/QWoVyZJ>

## Tasks

### Property – display: flex;



*For full screenshot check the screenshot.png file in the folder for the task*

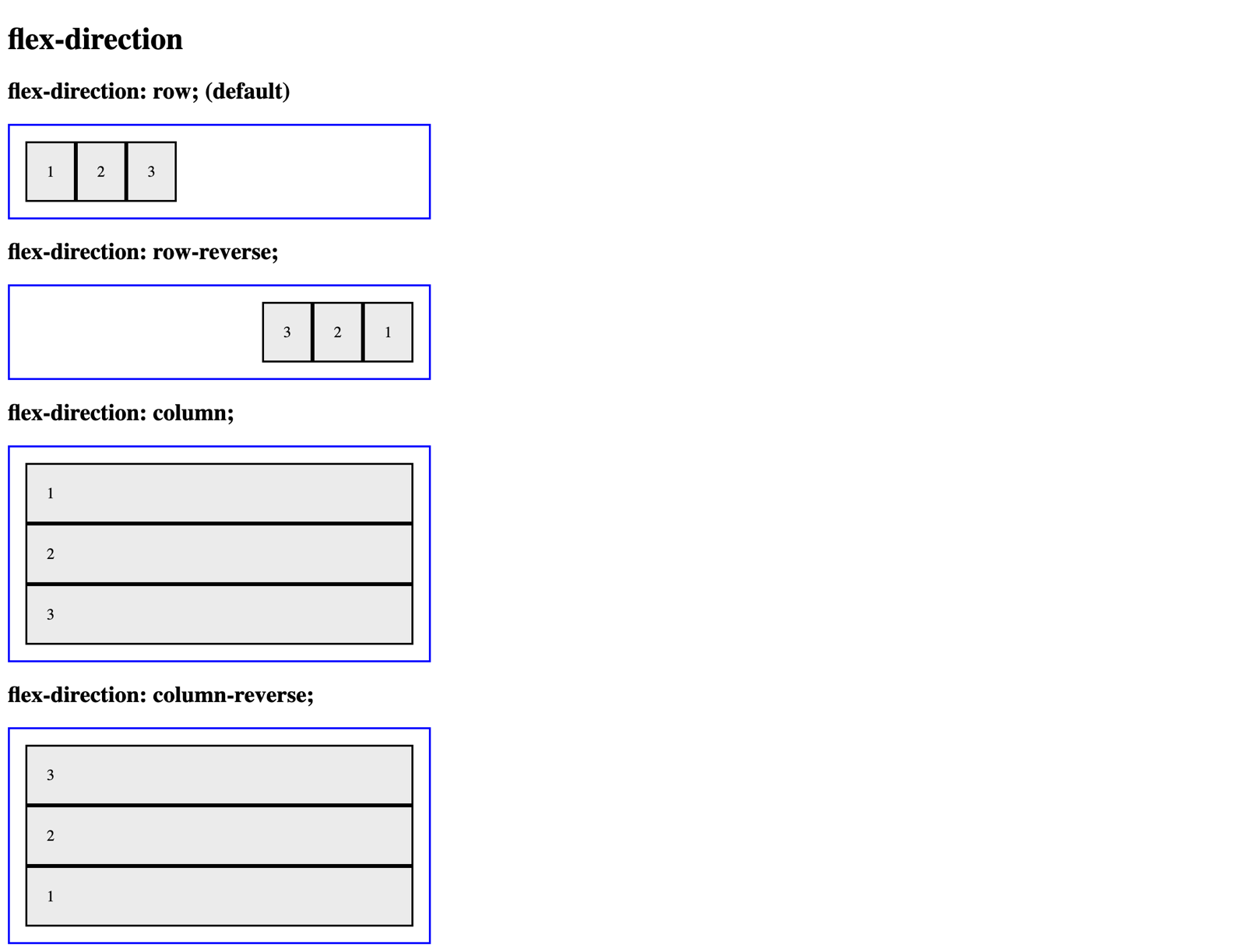
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **display-flex.html**
   * Create a **CSS** file named **display-flex.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create a **<div>** element with a class – "**flex-container**"
   * Create 3 <div> elements inside the .flex-container element with a class of "**flex-item**"
   * Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Add the following styles the .flex-container element:
     1. Width – 400px
     2. Padding – 1em
     3. Border – 2px solid blue
     4. Display – flex
   * Add the following styles to the .flex-item elements inside the .flex-container element:
     1. Background - #eee
     2. Padding – 20px
     3. Border – 2px solid #000

### Property – flex-direction



*For full screenshot check the screenshot.png file in the folder for the task*

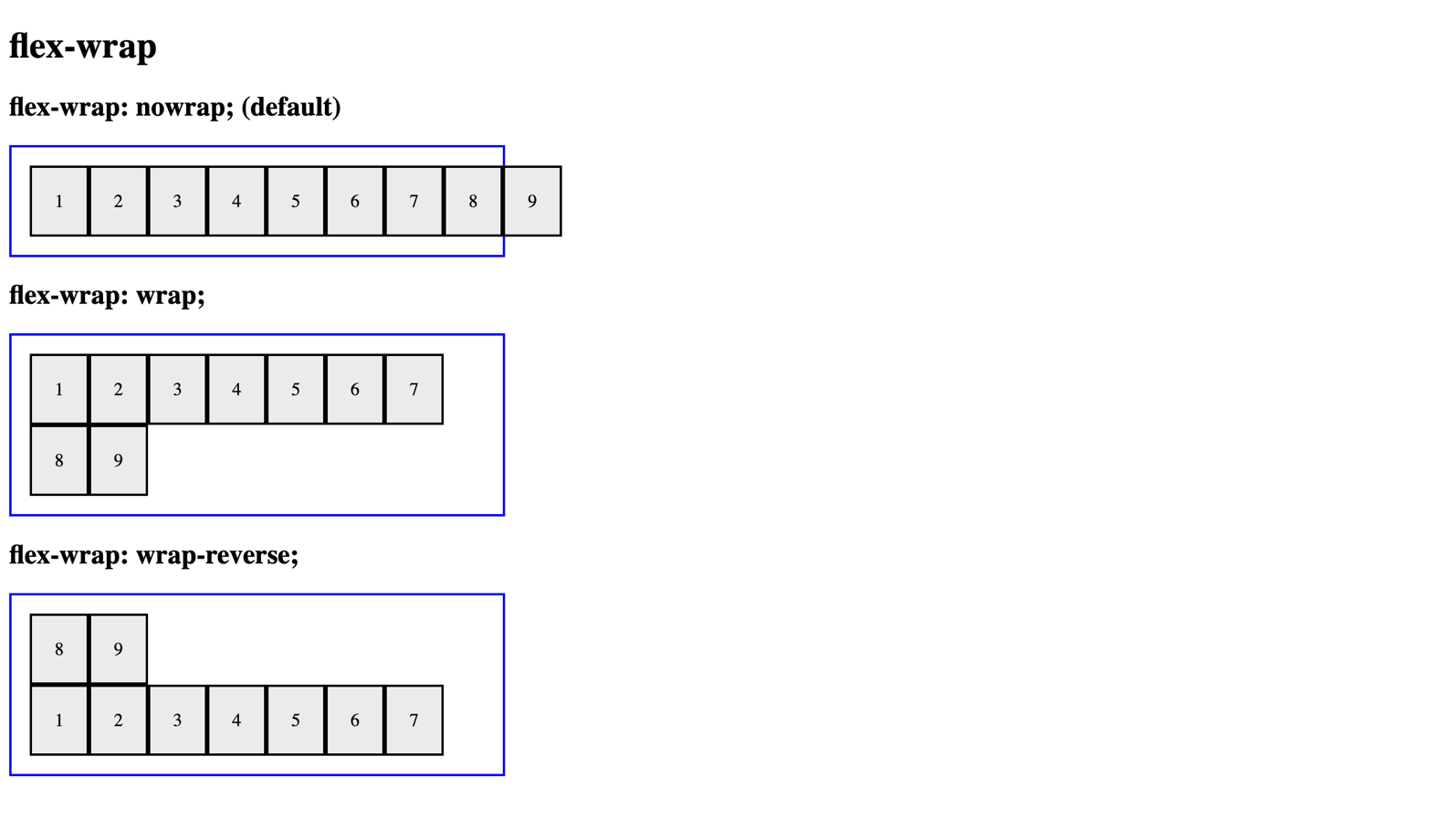
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **flex-direction.html**
   * Create a **CSS** file named **flex-direction.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 4 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item"**
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Create a new class **"flex-direction-row**" with the following styles
     1. Flex-direction – row
   * Create a new class **"flex-direction-row-reverse"** with the following styles
     1. Flex-direction – row-reverse
   * Create a new class **"flex-direction-column"** with the following styles
     1. Flex-direction – column
   * Create a new class **"flex-direction-column-reverse"** with the following styles
     1. Flex-direction – column-reverse
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property – flex-wrap



*For full screenshot check the screenshot.png file in the folder for the task*

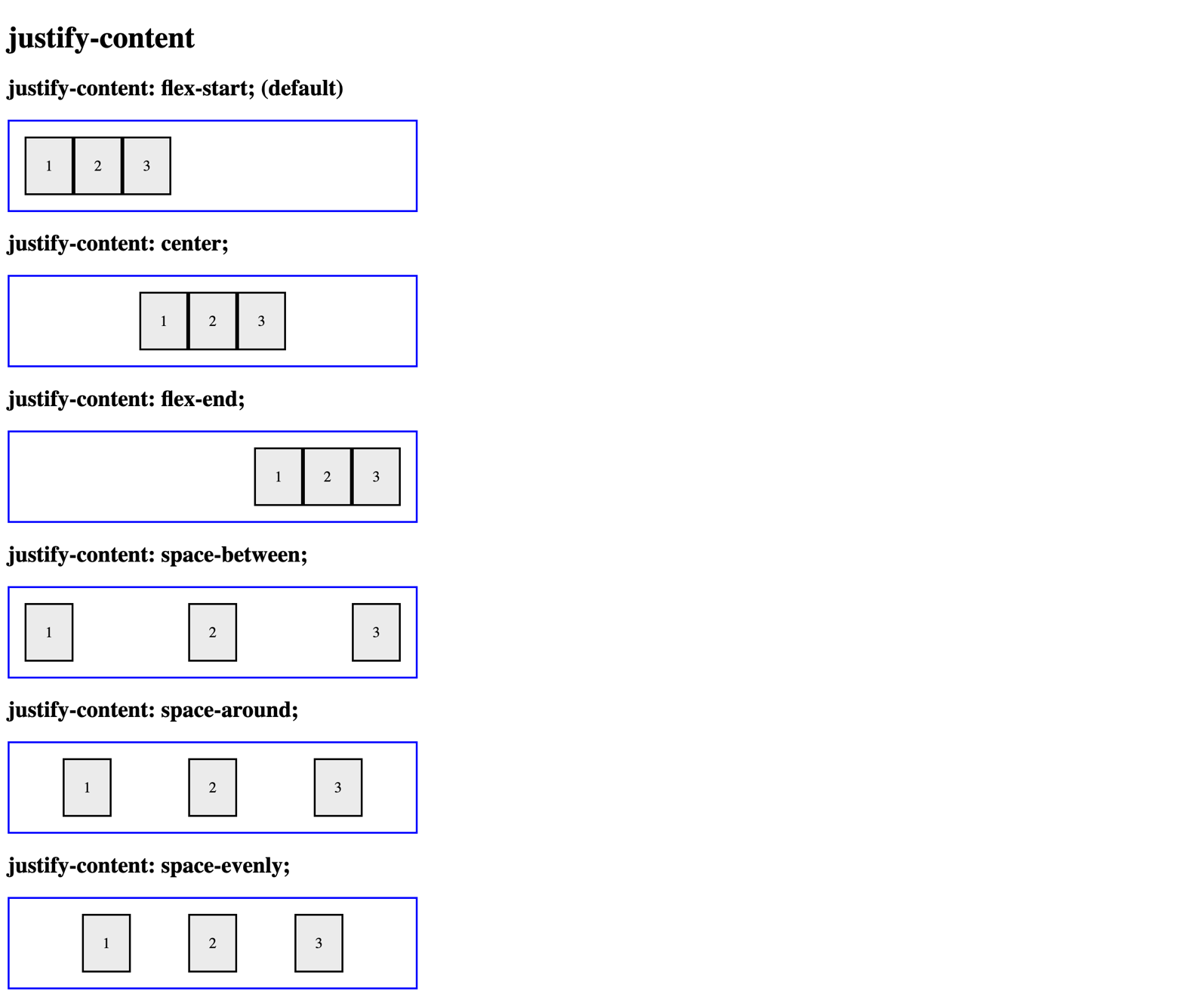
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **flex-wrap.html**
   * Create a **CSS** file named **flex-wrap.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 3 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Create a new class **"flex-wrap-nowrap"** with the following styles
     1. Flex-wrap – nowrap
   * Create a new class **"flex-wrap-wrap"** with the following styles
     1. Flex-wrap – wrap
   * Create a new class **"flex-wrap-wrap-reverse"** with the following styles
     1. Flex-wrap – wrap-reverse
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property – justify-content



*For full screenshot check the screenshot.png file in the folder for the task*

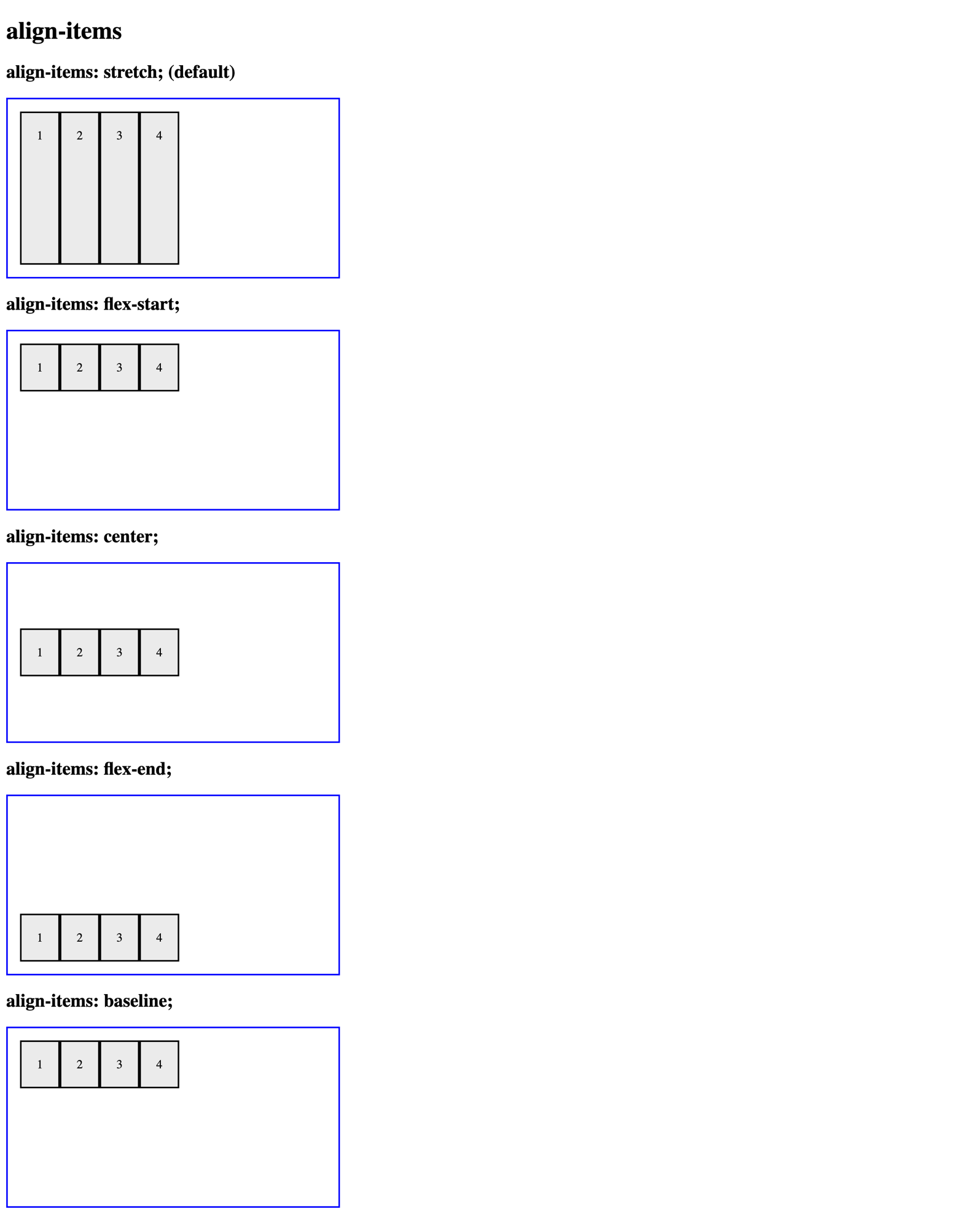
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **justify-content.html**
   * Create a **CSS** file named **justify-content.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 6 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Create a new class **"justify-content-flex-start"** with the following styles
     1. justify-content – flex-start
   * Create a new class **"justify-content-center"** with the following styles
     1. justify-content – center
   * Create a new class **"justify-content-flex-end"** with the following styles
     1. justify-content – flex-end
   * Create a new class **"justify-content-space-between"** with the following styles
     1. justify-content – space-between
   * Create a new class **"justify-content-space-around"** with the following styles
     1. justify-content – space-around
   * Create a new class **"justify-content-space-evenly"** with the following styles
     1. justify-content – space-evenly
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property – align-items



*For full screenshot check the screenshot.png file in the folder for the task*

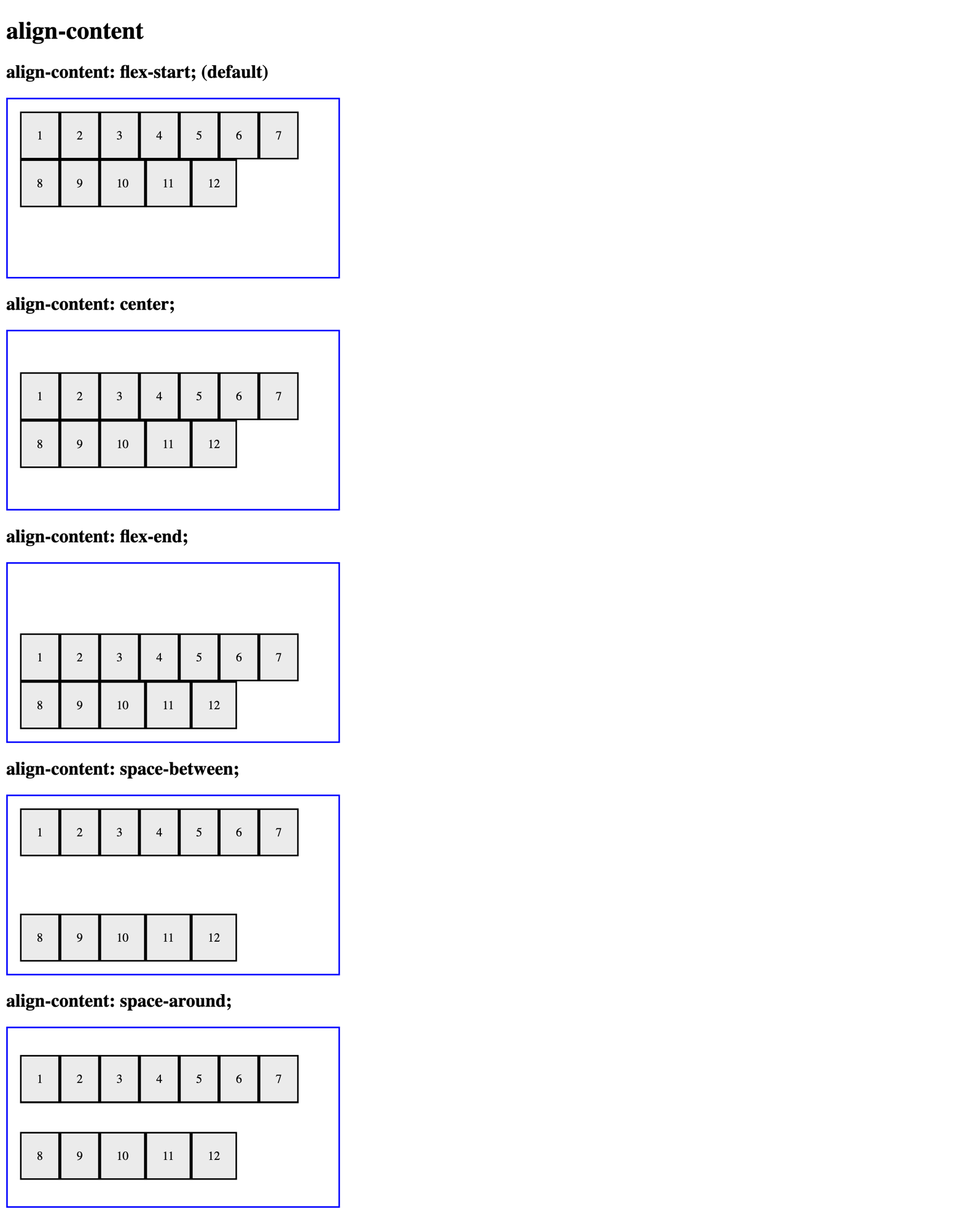
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **align-items.html**
   * Create a **CSS** file named **align-items.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 5 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. height - 200px
   * Create a new class **"align-items-****stretch"** with the following styles
     1. align-items – stretch
   * Create a new class "**align-items-flex-start"** with the following styles
     1. align-items – flex-start
   * Create a new class **"align-items-center"** with the following styles
     1. align-items – center
   * Create a new class **"align-items-flex-end"** with the following styles
     1. align-items – flex-end
   * Create a new class **"align-items-baseline"** with the following styles
     1. align-items – baseline
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Align Content

Use the skeleton is in resources to achieve the following view: 

*For full screenshot check the screenshot.png file in the folder for the task*

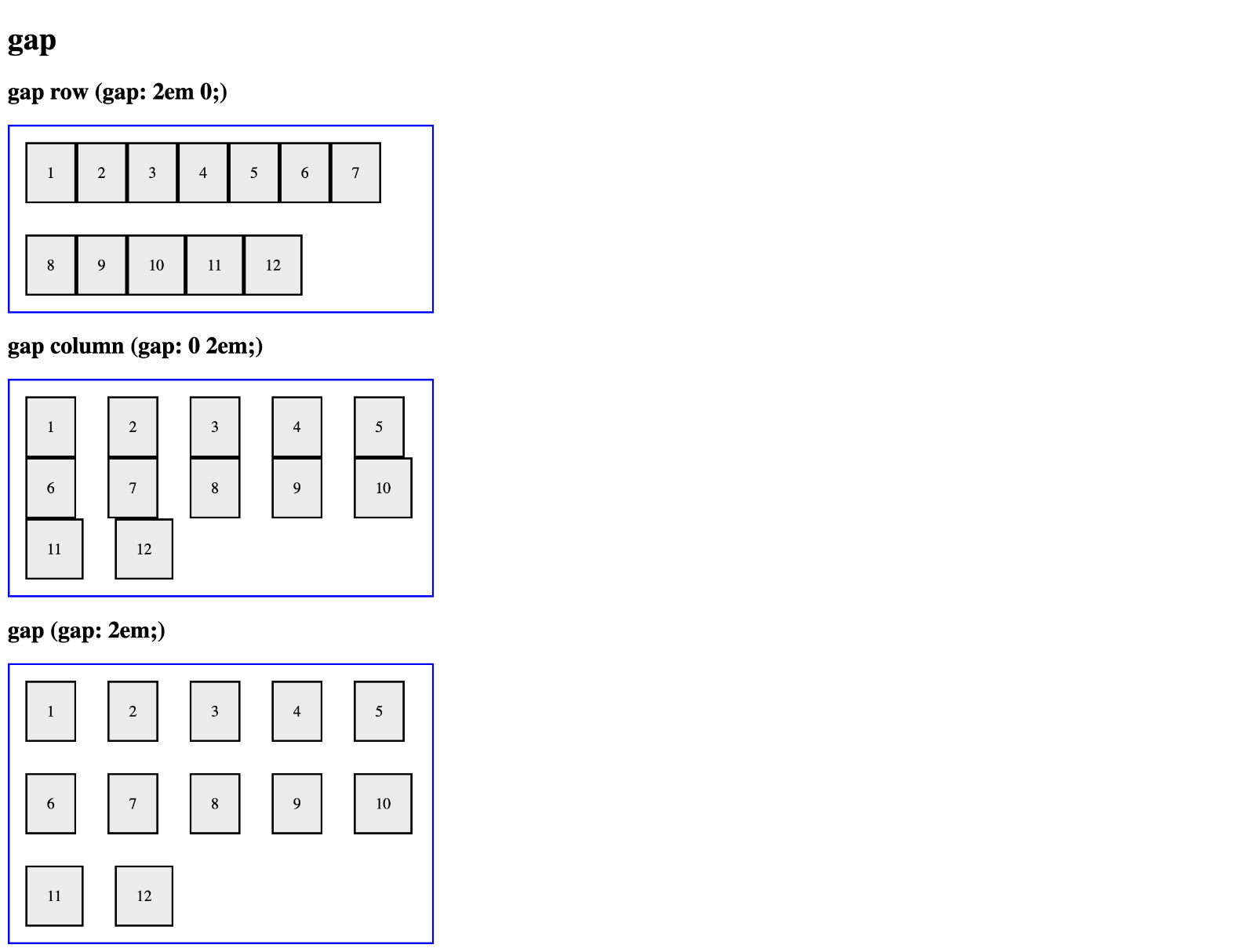
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **align-content.html**
   * Create a **CSS** file named **align-content.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 5 sets of **<h2>** element with a **<div>** element with a class – "**flex-container"** as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. height - 200px
     2. align-items - flex-start
     3. flex-wrap - wrap
   * Create a new class **"align-content-flex-start"** with the following styles
     1. align-content – flex-start
   * Create a new class **"align-content-center"** with the following styles
     1. align-content – center
   * Create a new class **"align-content-flex-end"** with the following styles
     1. align-content – flex-end
   * Create a new class **"align-content-****space-between"** with the following styles
     1. align-content – space-between
   * Create a new class **"align-content-space-between"** with the following styles
     1. align-content – space-around
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Gap



*For full screenshot check the screenshot.png file in the folder for the task*

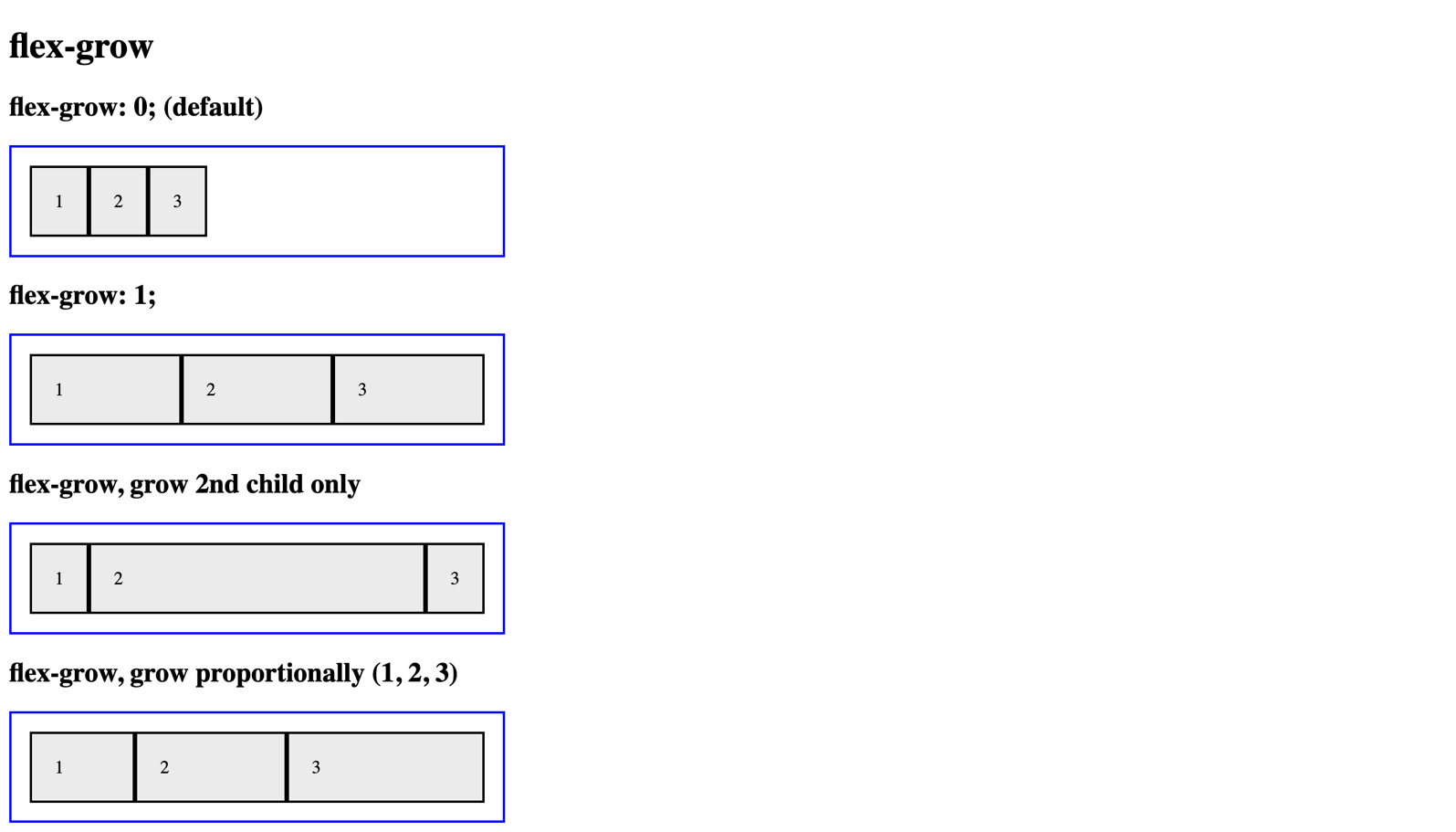
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **gap.html**
   * Create a **CSS** file named **gap.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 3 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. align-items – flex-start
     2. flex-wrap - wrap
   * Create a new class **"gap-row"** with the following styles
     1. gap - 2em 0;
   * Create a new class **"gap-column"** with the following styles
     1. gap - 0 2em;
   * Create a new class **"gap"** with the following styles
     1. gap - 2em;
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Flex Grow



*For full screenshot check the screenshot.png file in the folder for the task*

#### Objective:

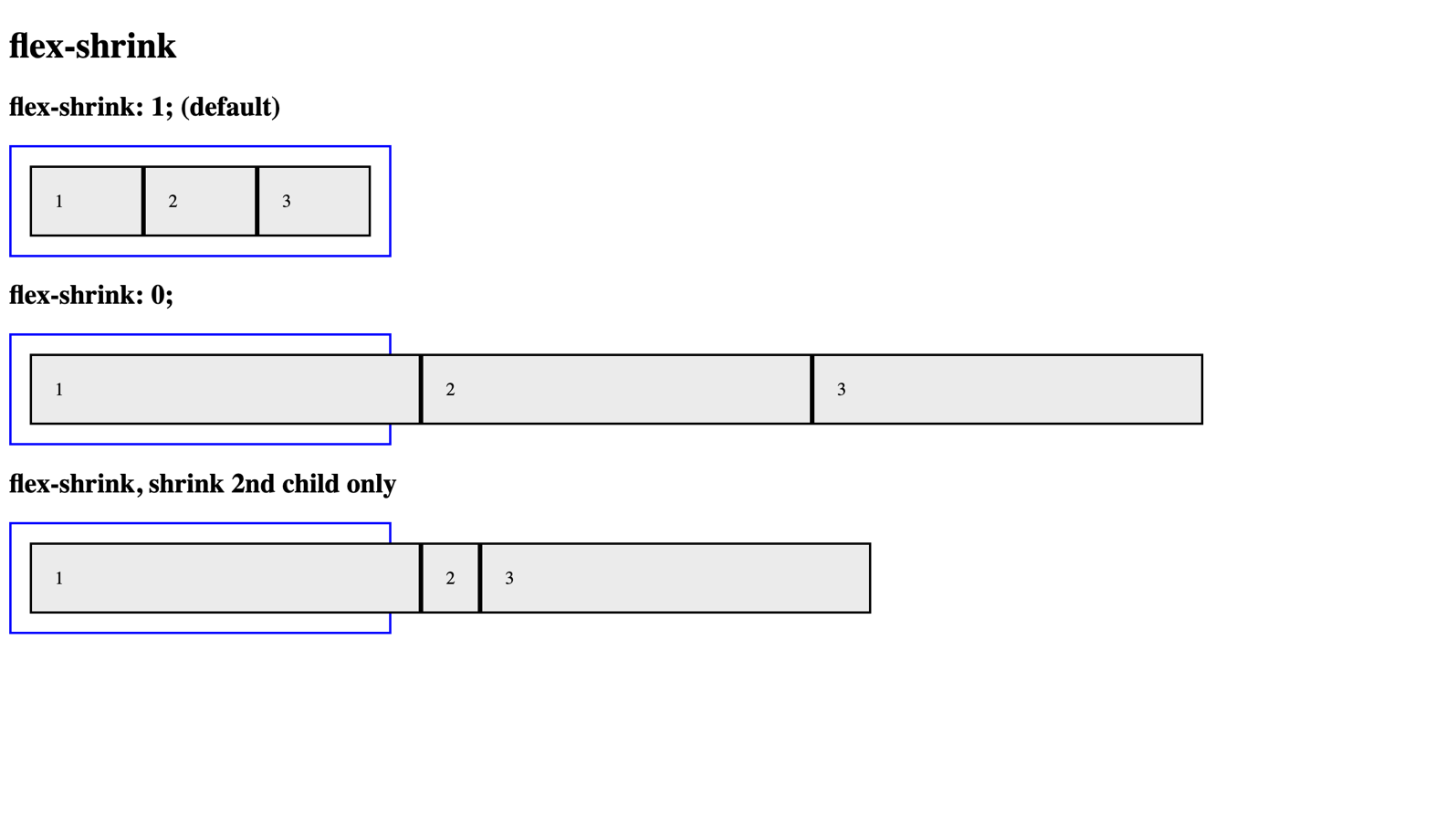
Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **flex-grow.html**
   * Create a **CSS** file named **flex-grow.css**

1. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 4 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
2. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. align-items – flex-start
     2. flex-wrap - wrap
   * Create a new class **"flex-grow-one"** and use it to style all .flex-item elements inside it with the following styles
     1. flex-grow - 1;
   * Create a new class **"flex-grow-2nd-only"** and use it to style the 2nd .flex-item element inside it with the following styles
     1. flex-grow - 1;
   * Create a new class **"flex-grow-proportional"** and use it to style each .flex-item element inside it with the following styles
     1. Element 1
        1. flex-grow - 1;
     2. Element 2
        1. flex-grow - 2;
     3. Element 3
        1. flex-grow - 3;
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Flex Shrink



*For full screenshot check the screenshot.png file in the folder for the task*

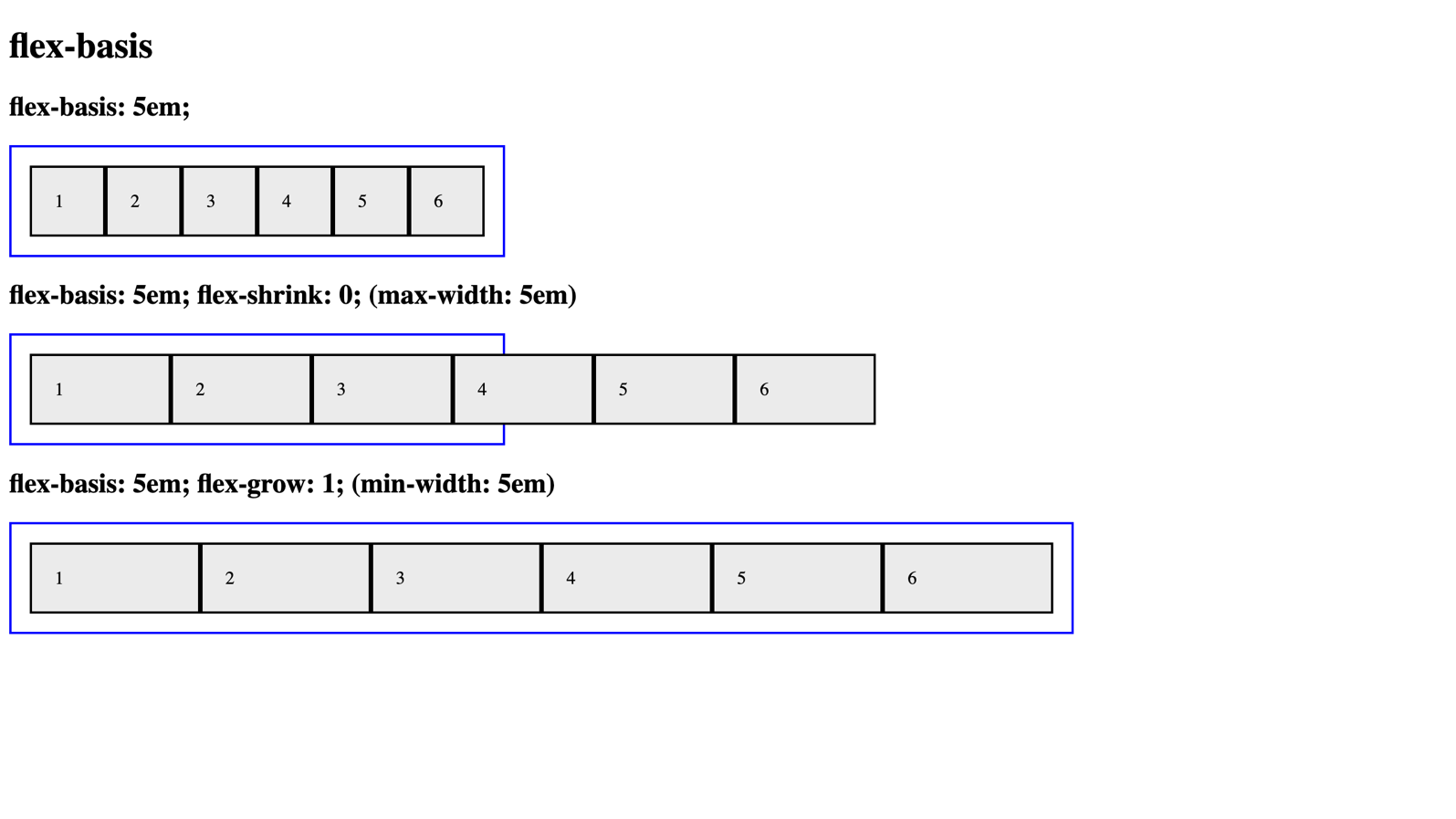
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **flex-shrink.html**
   * Create a **CSS** file named **flex-shrink.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 3 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-item class:
     1. flex-basis – 300px
   * Add the following styles to the .flex-container class:
     1. align-items – flex-start
     2. width – 300px
   * Create a new class **"flex-shrink-zero"** and use it to style all .flex-item elements inside it with the following styles:
     1. flex-shrink - 0;
   * Create a new class **"flex-shrink-2nd-only"** and use it to style all .flex-item element inside it with the following styles:
     1. flex-shrink - 0;
   * Use the class **"flex-shrink-2nd-only"** to style the 2nd .flex-item element inside it with the following styles:
     1. flex-shrink - 1;
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Flex Basis



*For full screenshot check the screenshot.png file in the folder for the task*

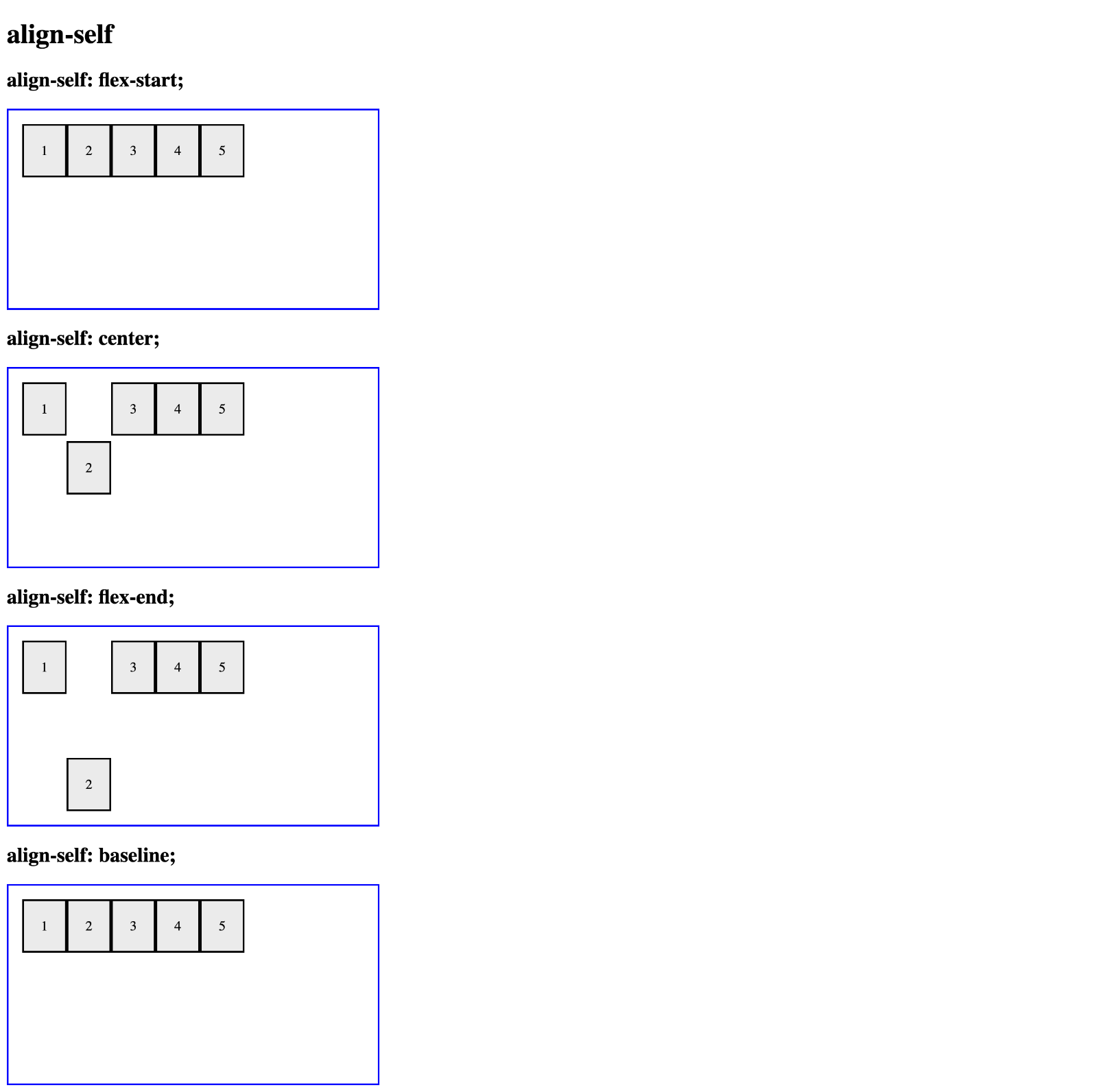
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **flex-basis.html**
   * Create a **CSS** file named **flex-basis.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 3 sets of **<h2>** element with a **<div>** element with a class – "**flex-container"** as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Create a new class **"flex-basis-5em"** and use it to style all .flex-item elements inside it with the following styles:
     1. flex-basis - 5em;
   * Create a new class **"flex-shrink-zero"** and use it to style all .flex-item element inside it with the following styles:
     1. flex-shrink - 0;
   * Create a new class **"flex-grow-one"** with the following styles:
     1. Width - 900px;
   * Use the class **"flex-grow-one"** it to style all .flex-item element inside it with the following styles:
     1. flex-grow - 1;
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Align Self



*For full screenshot check the screenshot.png file in the folder for the task*

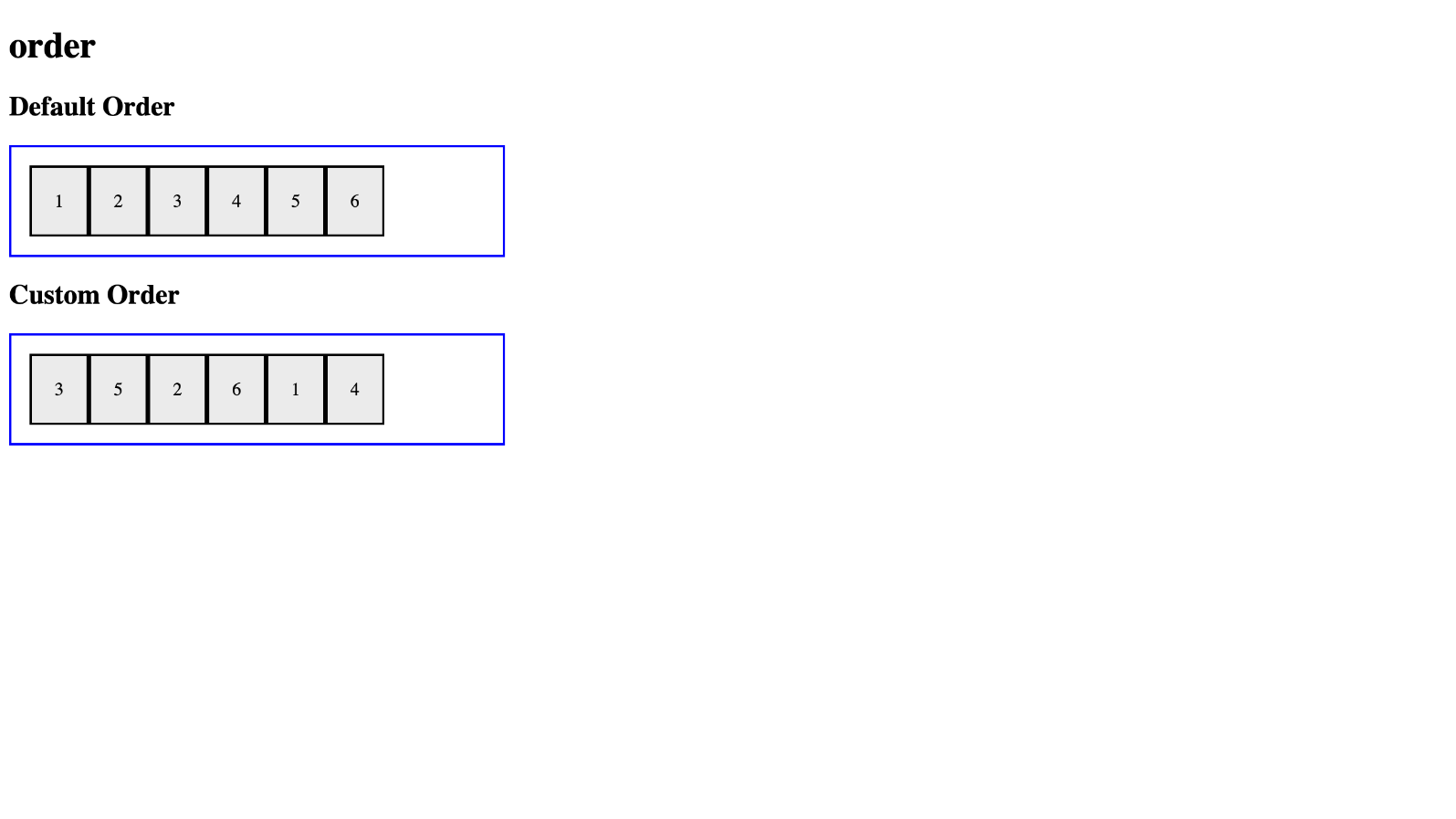
#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an **HTML** document named **align-self.html**
   * Create a **CSS** file named **align-self.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 4 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. align-items – flex-start
   * Create a new class **"align-self-flex-start"** and use it to style the 2nd .flex-item element inside it with the following styles:
     1. align-self - flex-start;
   * Create a new class **"align-self-center"** and use it to style the 2nd .flex-item element inside it with the following styles:
     1. align-self - center;
   * Create a new class **"align-self-flex-end"** and use it to style the 2nd .flex-item element inside it with the following styles:
     1. align-self - flex-end;
   * Create a new class **"align-self-baseline"** and use it to style the 2nd .flex-item element inside it with the following styles:
     1. align-self - baseline;
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot

### Property: Order



*For full screenshot check the screenshot.png file in the folder for the task*

#### Objective:

Create a webpage that looks like the screenshot above.

#### Requirements:

1. **File Setup**:
   * Create an HTML document named **order.html**
   * Create a CSS file named **order.css**
2. **Element creation**
   * Add title of the page in an **<h1>** element
   * Create 2 sets of **<h2>** element with a **<div>** element with a class – "**flex-container**" as shown in the screenshot
     1. Create **<div>** elements inside the .flex-container element with a class of "**flex-item**"
     2. Add text to the .flex-item elements as shown on the screenshot
3. **Styling**
   * Copy the styles from Task 1
   * Add the following styles to the .flex-container class:
     1. align-items – flex-start
   * Create a new class “custom-order” and use it to style each individual .flex-item element inside it with the following styles:
     1. Item 1
        1. Order – 5
     2. Item 2
        1. Order – 3
     3. Item 3
        1. Order – 1
     4. Item 4
        1. Order – 6
     5. Item 5
        1. Order – 2
     6. Item 6
        1. Order – 4
   * Apply the classes to the appropriate .flex-container elements to achieve the results shown in the screenshot