CSS Box Model

In fixed-width layouts, all widths matter! On Day 14, we created a new fixed-width layout framework with the following rule:

```
.container {
      width: 960px;
      ...
}
```

This gives us a 'budget' for the width of our layout. With this `.container` class, all of the parts of all of our boxes must **add up to 960 pixels** altogether.

Box Model Components

When measuring your layout, you must consider the following components (and corresponding CSS properties) of **each element** (each box) on every row:

```
margin-left + margin-right
border-left + border-right
padding-left + padding-right
width (of content)
```

Worksheet

For each of the following layouts:

1. Draw your **fixed-width** container. (Assume that it is 960 pixels wide.)
2. Draw your **layout** inside of the container.
3. Label all of the **measurements** for your diagram.
4. Determine the **widths** of each box in your layout.
5. Write the **CSS properties and values** for this layout.

Parent & Child Containers (in a Single Column Layout)

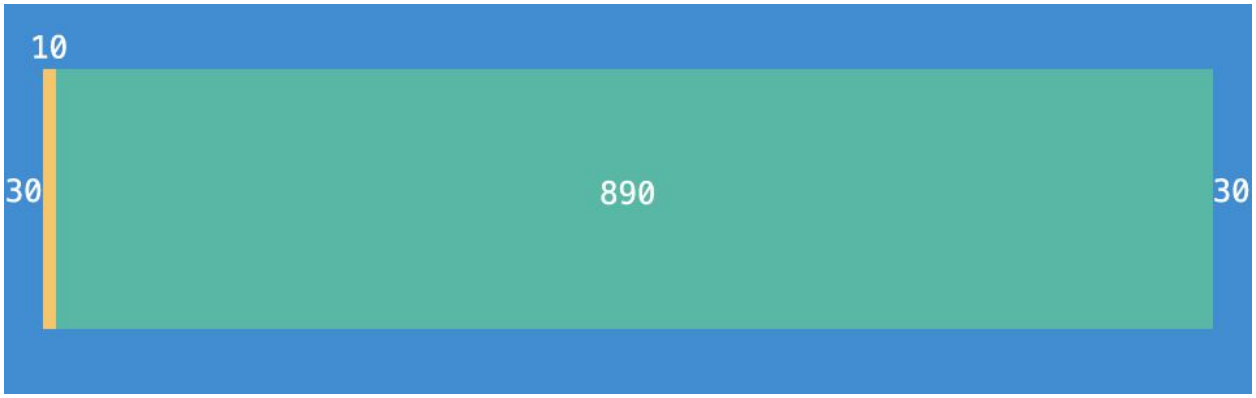
Given the following HTML:

```
<div class="container">
      <main>
            <section></section>
      </main>
</div>
```

… create a layout with the following features:

- ➔ 2 boxes (`<main>` and `<section>`) in a single column
- ➔ 30 pixels of space between the edges of `<main>` and `<section>`
- ➔ 10 pixels of border, only on the left side of `<section>`

Assume that there are no margins on the `<main>` element (i.e. it is flush against the `.container`).



`main {`
`  padding: 30px;`
`}`
	`section {`
	`  height: 300px;`
	`  width: 890px;`
	`  border-left: 10px;`
	`}`

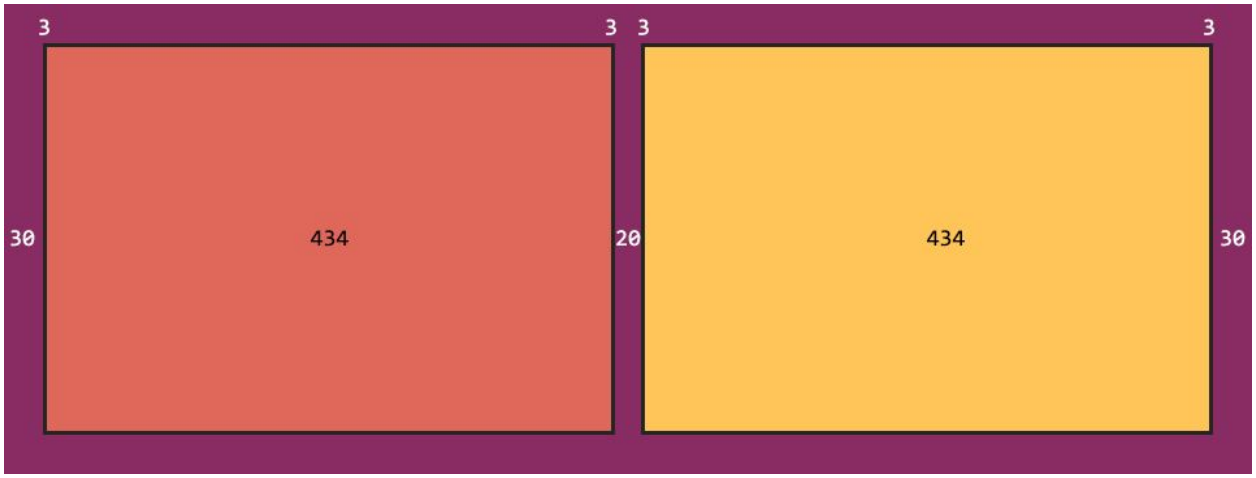
Two Column Layout

Given the following HTML:

```
<div class="container">
      <div class="flex">
            <div class="box-left"></div>
            <div class="box-right"></div>
      </div> <!-- end of .flex -->
</div> <!-- end of .container -->
```

Create a layout with the following features:

- ➔ 2 boxes of the same size (`.box-left` and `.box-right`), arranged side-by-side
- ➔ a border with a 3 pixel width on both boxes
- ➔ 20 pixels of space between the boxes
- ➔ 30 pixels of space between the boxes and the `.container`



`.flex {`
`  display: flex;`
`  padding: 30px;`
`}`
	`.box-left {`
	`  height: 300px;`
	`  border: 3px solid black;`
	`  margin-right: 20px;`
	`}`
		`.box-right {`
		`  height: 300px;`
		`  border: 3px solid black;`
		`}`

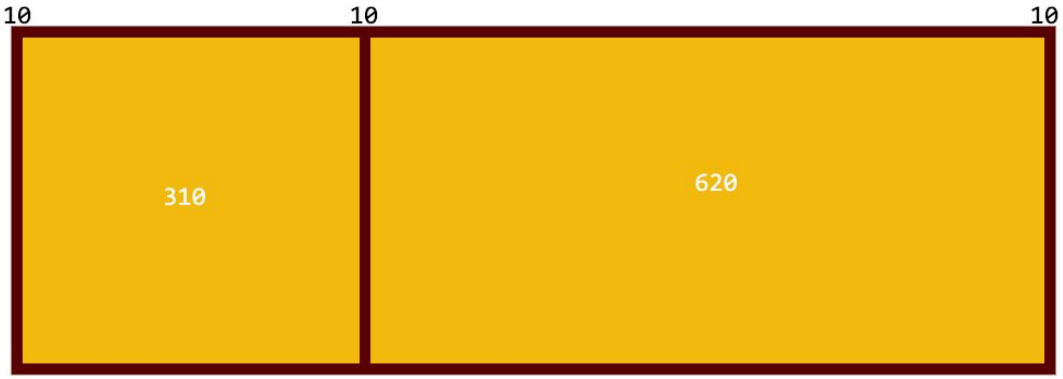
Uneven Two Column Layout

Given the following HTML:

```
<div class="container">
      <div class="flex">
            <div class="box-short"></div>
            <div class="box-long"></div>
      </div> <!-- end of .flex -->
</div> <!-- end of .container -->
```

Create a layout with the following features:

- ➔ 2 boxes (`.box-short` and `.box-long`), arranged side-by-side
- ➔ `.box-long` is twice the width of `.box-short`
- ➔ 10 pixels of space between either box
- ➔ 10 pixels of space between the edge of `.container` and the boxes



We can express this algebraically.

$$3(10) + x + 2x = 960$$
$$30 + 3x = 960$$
$$3x = 930$$
$$x = 310$$

If we let `x` represent `.box-short` and `2x` represent `.box-long`, then …

`.flex {`
`  display: flex;`
`  padding: 10px;`
`}`
	`.box-short {`
	`  height: 300px;`
	`  width: 310px;`
	`  margin-right: 10px;`
	`}`
		`.box-long {`
		`  height: 300px;`
		`  width: 620px;`
		`}`

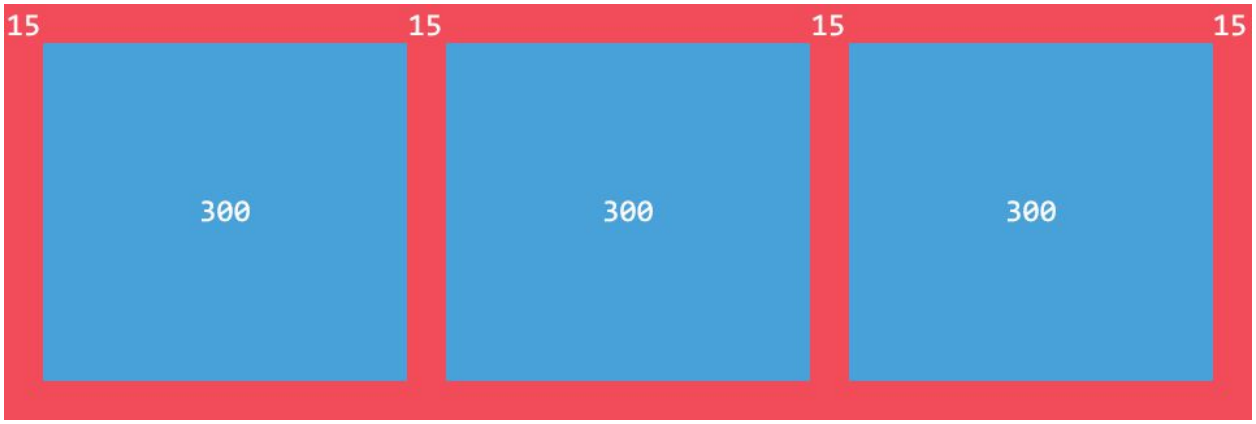
Three Column Layout

Given the following HTML:

```
<div class="container">
      <div class="flex">
            <div class="left"></div>
            <div class="middle"></div>
            <div class="right"></div>
      </div> <!-- end of .flex -->
</div> <!-- end of .container -->
```

Create a layout with the following features:

- ➔ 3 boxes of equal width
- ➔ 15 pixels of space between each box
- ➔ 15 pixels of space between the edge of the `.container` and the outermost boxes



`.flex {`
`  display: flex;`
`  padding: 15px;`
`}`
	`.left {`
	`  height: 300px;`
	`  width: 300px;`
	`  margin-right: 15px;`
	`}`

`.middle {`
`  height: 300px;`
`  width: 300px;`
`  margin-right: 15px;`
`}`
	`.right {`
	`  height: 300px;`
	`  width: 300px;`
	`}`