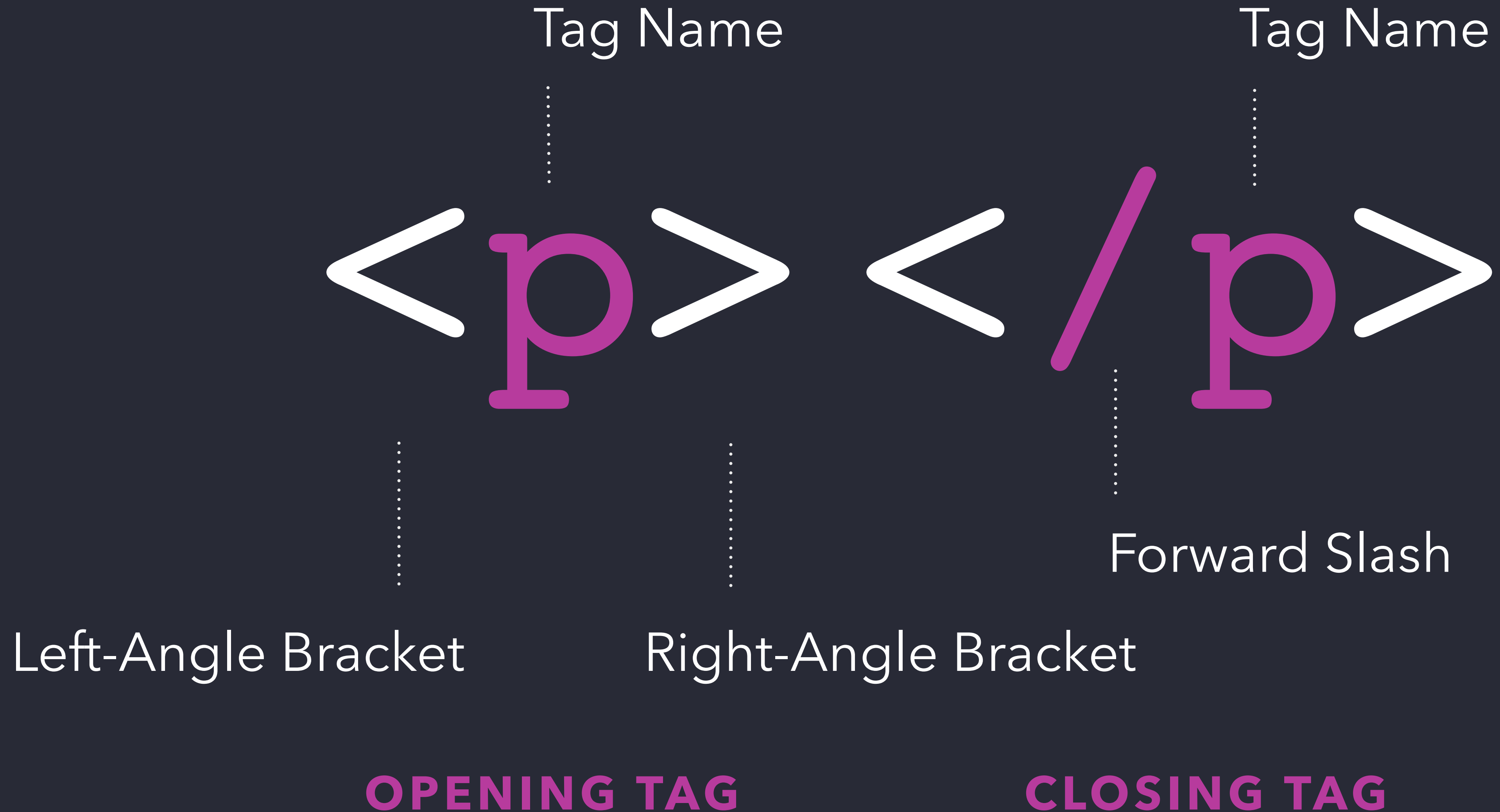


HTML & CSS REVIEW

HTML

HTML TAGS (ELEMENTS)



CONTENT GOES BETWEEN TAGS

```
<p>Lorem ipsum dolor.</p>
```

ATTRIBUTES TELL US MORE ABOUT AN HTML ELEMENT

Attribute Name

⋮

```
<a href="http://google.com">Google</a>
```

⋮

Attribute Value

HTML: PAGE STRUCTURE

CODE

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Document Title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is the Main Heading</h1>
```

```
<p>This text might be an introduction to the rest of the page.</p>
```

```
<h2>This is a Sub-Heading</h2>
```

```
<p>Many long articles have sub-headings to help you follow the structure.</p>
```

```
</body>
```

```
</html>
```

LINKS

ADDING LINKS

THE PAGE THE LINK
TAKES YOU TO



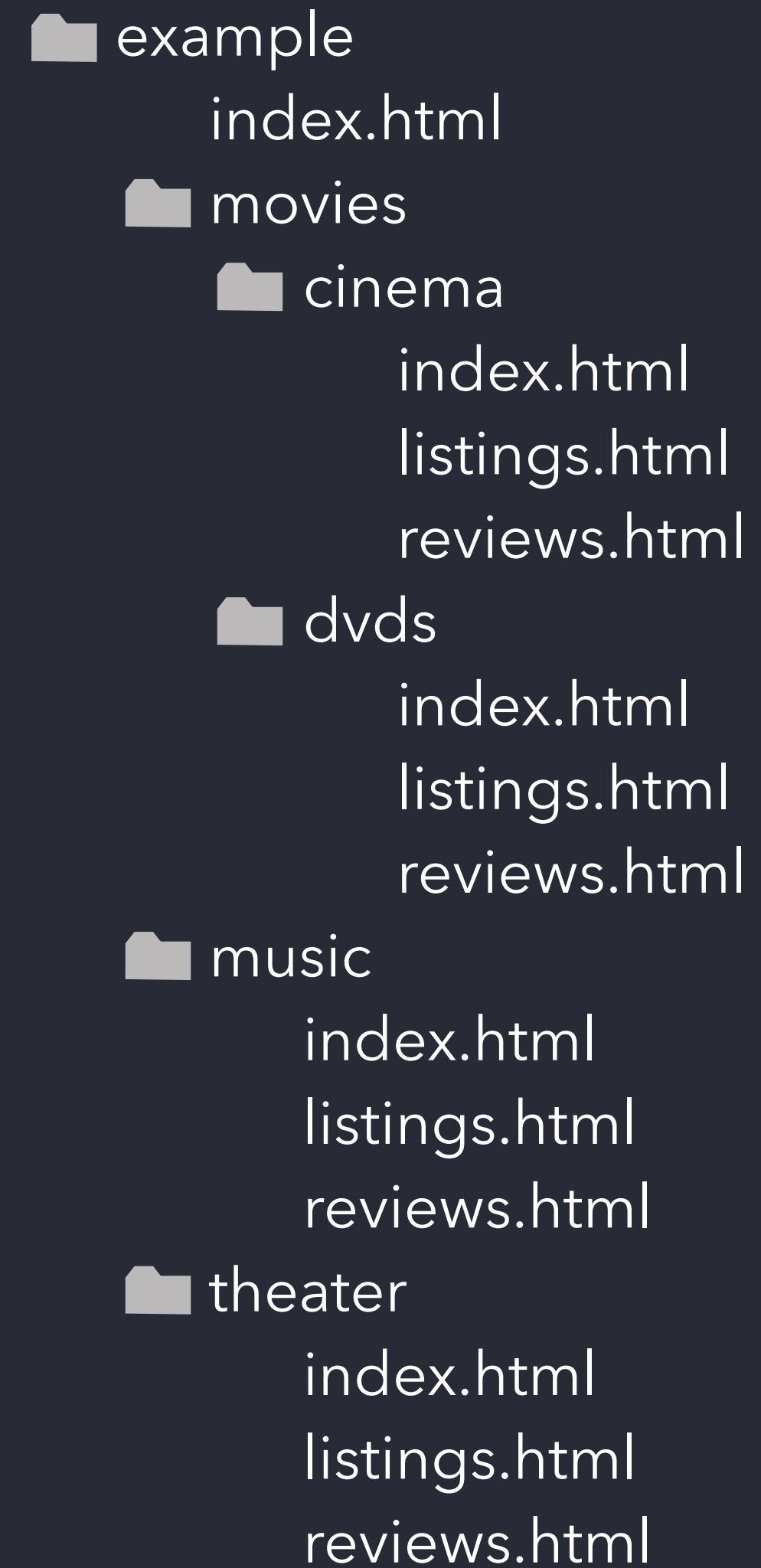
```
<a href="http://www.imdb.com">IMDB</a>
```



THE TEXT THE
USER CLICKS ON

DIRECTORY STRUCTURE

Root Folder



DIRECTORY STRUCTURE

Child



DIRECTORY STRUCTURE

Parent



DIRECTORY STRUCTURE

Grandchild



DIRECTORY STRUCTURE

Grandparent



RELATIVE URLS

Same folder

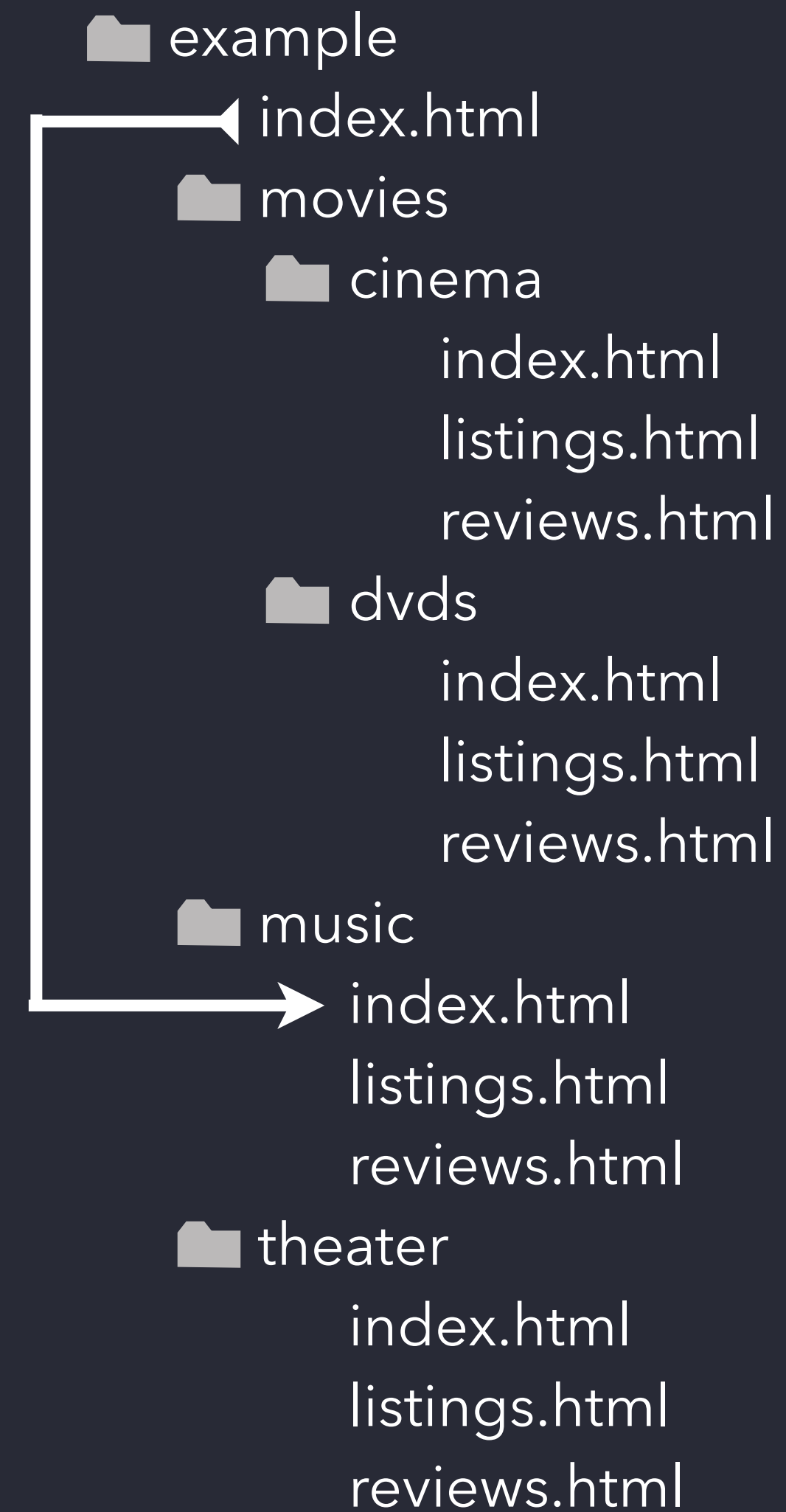
`reviews.html`



RELATIVE URLS

Child

`music/index.html`



RELATIVE URLS

Parent

`../index.html`



RELATIVE URLS

Grandchild

`movies/dvds/index.html`



RELATIVE URLS

Grandparent

`../../index.html`



<https://developer.mozilla.org/en-US/docs/Web/HTML/Reference>

CSS

CSS ASSOCIATES STYLE RULES WITH HTML ELEMENTS

```
p { font-family: Arial; }
```

Selector

Declaration

CSS PROPERTIES AFFECT HOW ELEMENTS ARE DISPLAYED

```
h1 {color: yellow;}
```

Property

Value

CSS PROPERTIES AFFECT HOW ELEMENTS ARE DISPLAYED

```
h1 {  
  color: yellow;  
  font-family: Arial;  
  font-size: 18px;  
}
```

Properties

Values

USING EXTERNAL CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" rel="stylesheet">
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of...</p>
  </body>
</html>
```

CODE

USING EXTERNAL CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" rel="stylesheet">
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of...</p>
  </body>
</html>
```

CODE

CSS SELECTORS

Universal

```
* {}
```

Type

```
h1, h2, h3 {}
```

Class

```
.note {}  
p.note {}
```

Descendent

```
p a {}
```

ID

```
#introduction {} /* NEVER use these */
```

CASCADE & INHERITANCE

CASCADE

Selectors **further down** a style sheet **override** the same selectors **higher** in the style sheet

CASCADE

```
h1 {  
  color: green;  
}
```

```
h1 {  
  color: red;  
}
```

CODE

CASCADE

```
h1 {  
  color: green;  
}
```

CODE

```
h1 {  
  color: red; /* All h1 will be red, overriding green set above */  
}
```

INHERITANCE

Some properties **inherit** styles from **parent elements**

INHERITANCE

```
body {  
  font-family: Arial;  
  color: #333;  
  padding: 10px;  
}
```

```
h1 { ... }
```

```
.page { ... }
```

CODE

INHERITANCE

```
body {  
  font-family: Arial; /* Inherited by children */  
  color: #333; /* Inherited by children */  
  padding: 10px;  
}
```

```
h1 { ... }
```

```
.page { ... }
```

CODE

INHERITANCE

```
body {  
  font-family: Arial;  
  color: #333;  
  padding: 10px;  
}
```

```
h1 {  
  color: #acd123; /* Override color set on parent (body) */  
}
```

```
.page { ... }
```

CODE

INHERITANCE

```
body {  
  font-family: Arial;  
  color: #333;  
  padding: 10px; /* Not inherited by children */  
}  
  
h1 {  
  color: #acd123; /* Override color set on parent (body) */  
}  
  
.page {  
  padding: inherit; /* Force inheritance from parent (body) */  
}
```

CODE

SPECIFICITY

SPECIFICITY

Selectors that are **more specific** will **override** selectors that are **less specific**
(regardless of placement in stylesheet)

CALCULATING SPECIFICITY

A weight is applied to a CSS selector

Weight is determined by the number of **each selector types** in the selector:

0. Type selectors (e.g. `h1`)
1. Class selectors (e.g. `.example`)
2. ID selectors (e.g. `#example`)

<https://specificity.keegan.st>

CALCULATING SPECIFICITY

```
h1 { ... }  
0 IDs      0 classes    1 element    = 1
```

```
body header h1 { ... }  
0 IDs      0 classes    3 elements   = 3
```

```
.primary { ... }  
0 IDs      1 class       0 elements   = 10
```

```
h1.primary { ... }  
0 IDs      1 class       1 element    = 11
```

```
.hero h1.primary { ... }  
0 IDs      2 classes    1 element    = 21
```

```
#primary-header { ... }  
1 ID       0 classes    0 elements   = 100
```

CODE

CSS BOX MODEL

BUILDING BLOCKS

BLOCK LEVEL

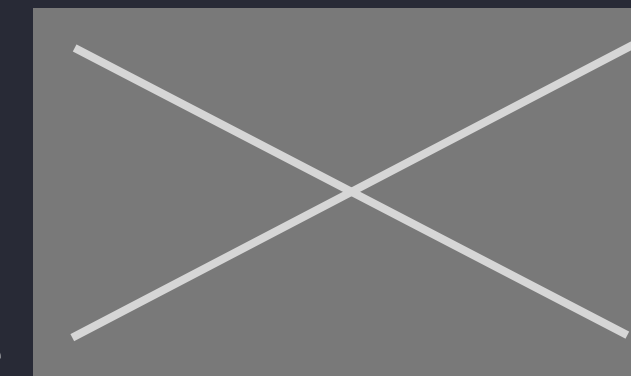
LOREM IPSUM

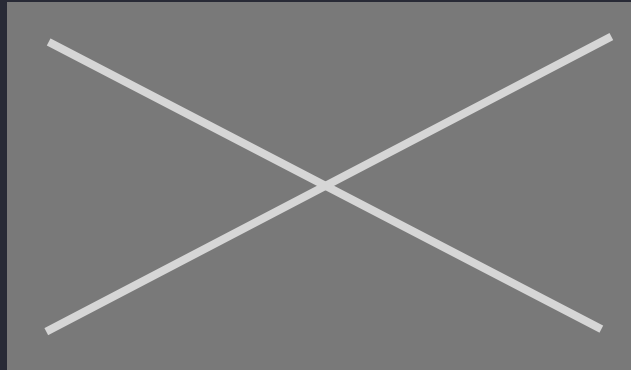
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam sodales pretium ipsum. Etiam ut enim augue. Etiam mi tortor, pulvinar at dictum faucibus, mollis eget nunc. Morbi justo velit, rutrum vel placerat in, adipiscing vitae sapien.

- Duis in erat neque.
- Pellentesque habitant morbi
- Praesent ac condimentum neque

INLINE

*Lorem ipsum dolor sit amet, [consectetur adipiscing](#) elit. Nullam sodales **pretium ipsum**. Etiam ut enim augue. Etiam mi tortor, pulvinar at dictum faucibus, mollis eget nunc. Morbi justo velit, rutrum vel placerat in, adipiscing.*



Suspendisse  potenti. Duis in erat neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

THINKING INSIDE THE BOX

The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense planting and a mixture of ornamental and edible plants.

The Cottage Garden originated in England and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained English estate gardens.

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

WIDTH & HEIGHT

Hello | HEIGHT

WIDTH

A diagram illustrating the width and height of the word "Hello". The word "Hello" is written in white. A red horizontal bar is positioned below the word, spanning its entire width, with the word "WIDTH" in red text centered below it. A green vertical bar is positioned to the right of the word, spanning its entire height, with the word "HEIGHT" in green text to its right.

BORDER, MARGIN AND PADDING



PADDING

BORDER

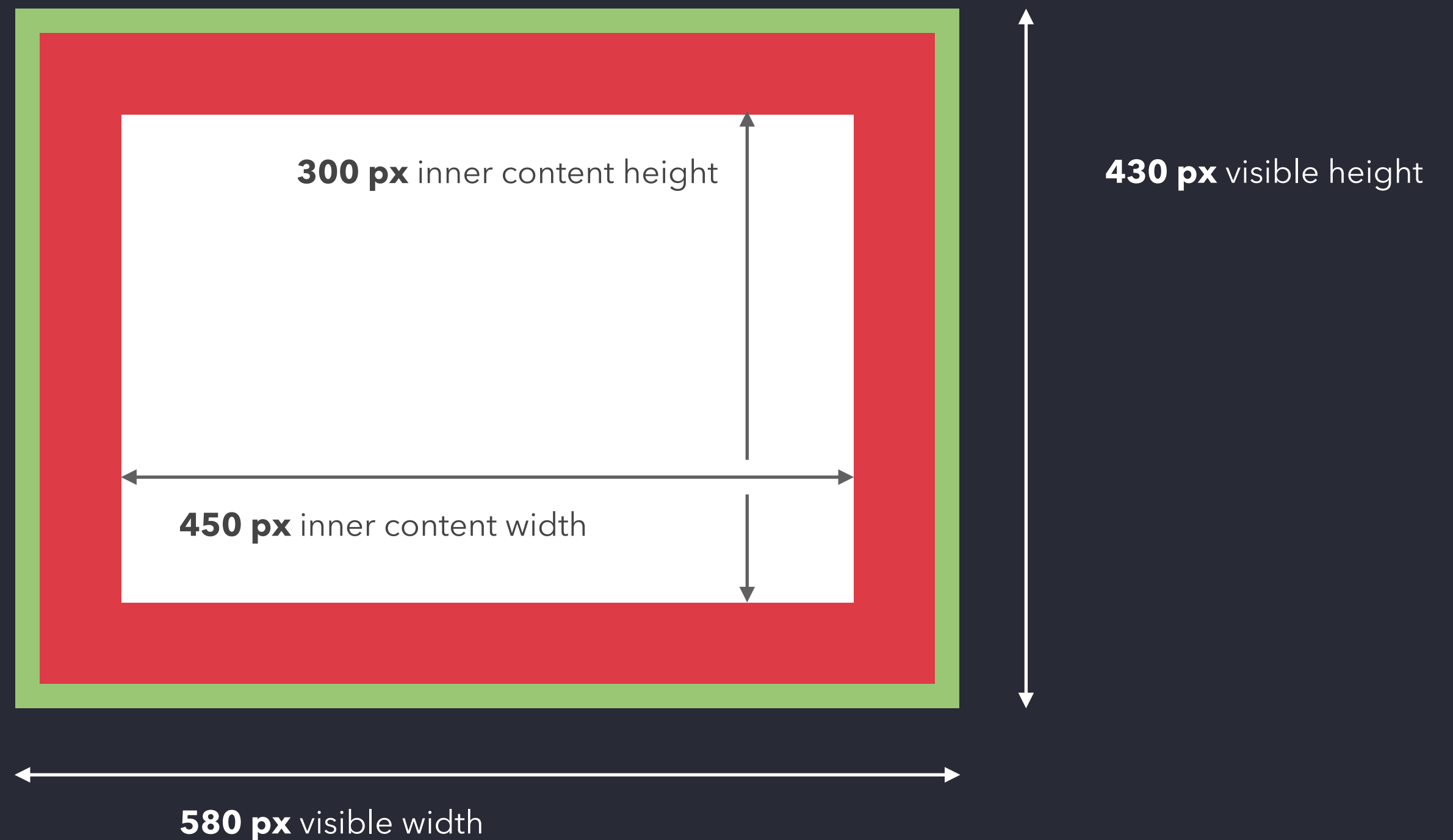
MARGIN

BOX SIZING

By default padding and border are **added** to the width of a box

$\text{width} + \text{padding} + \text{border} = \text{actual visual width of box}$

```
div {  
  width: 450px;  
  height: 300px;  
  padding: 50px;  
  border: 15px solid red;  
}
```



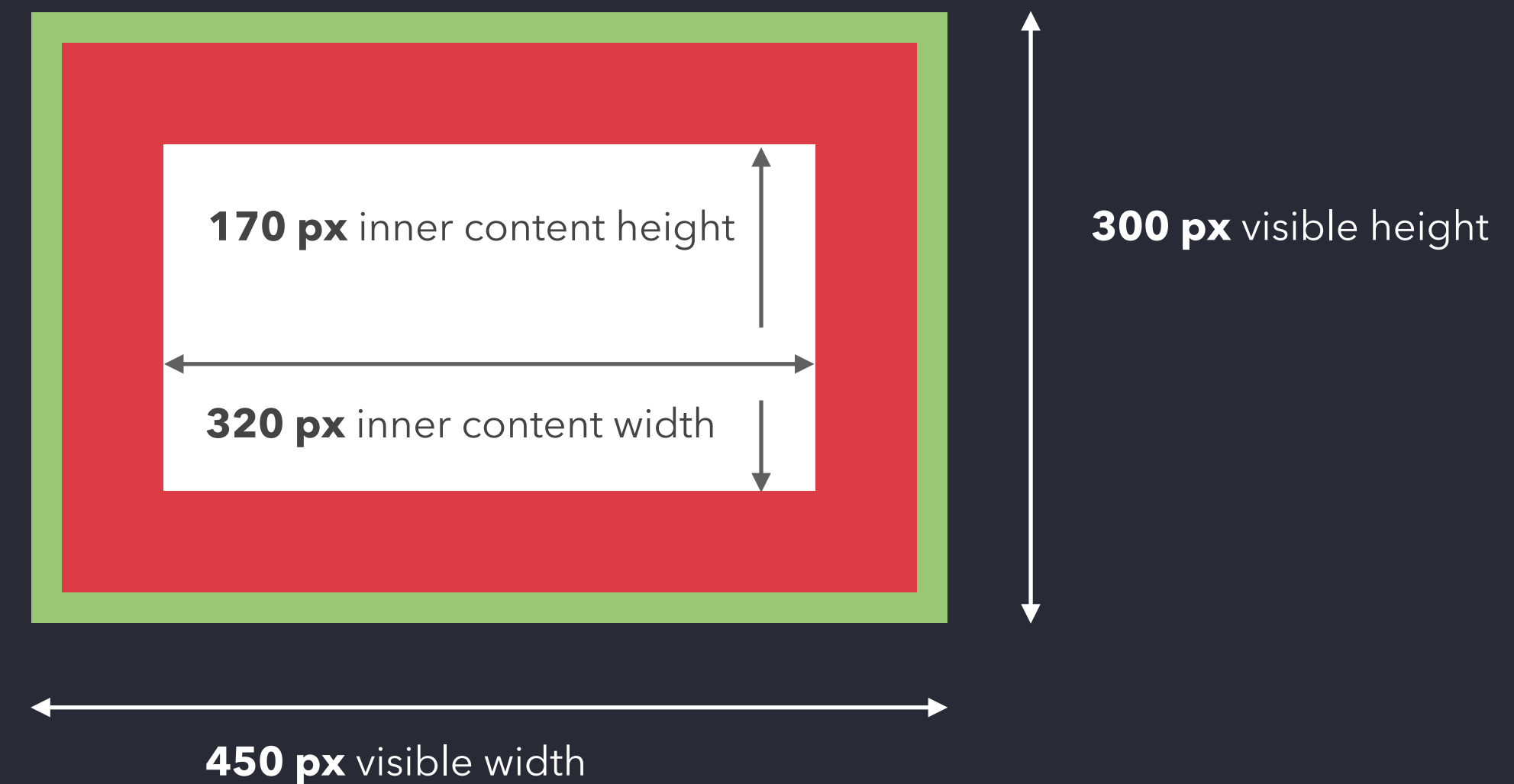
* white box is where text content shows

BOX SIZING: BORDER-BOX

`border-box`: padding and border are **included** in the width of a box

Therefore reducing the inner dimensions of the box

```
div {  
  width: 450px;  
  height: 300px;  
  padding: 50px;  
  border: 15px solid red;  
  box-sizing: border-box;  
}
```



* white box is where text content shows

BOX SIZING

This becomes especially important when mixing percentage based widths with pixel based padding.

```
.div-1 {  
  width: 100%; /* Visual width is 100% + 100px = >100% */  
  padding: 50px;  
}
```

```
.div-2 {  
  width: 100%; /* Inner content width is 100% - 100px */  
  padding: 50px;  
  box-sizing: border-box;  
}
```

FLEXBOX

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

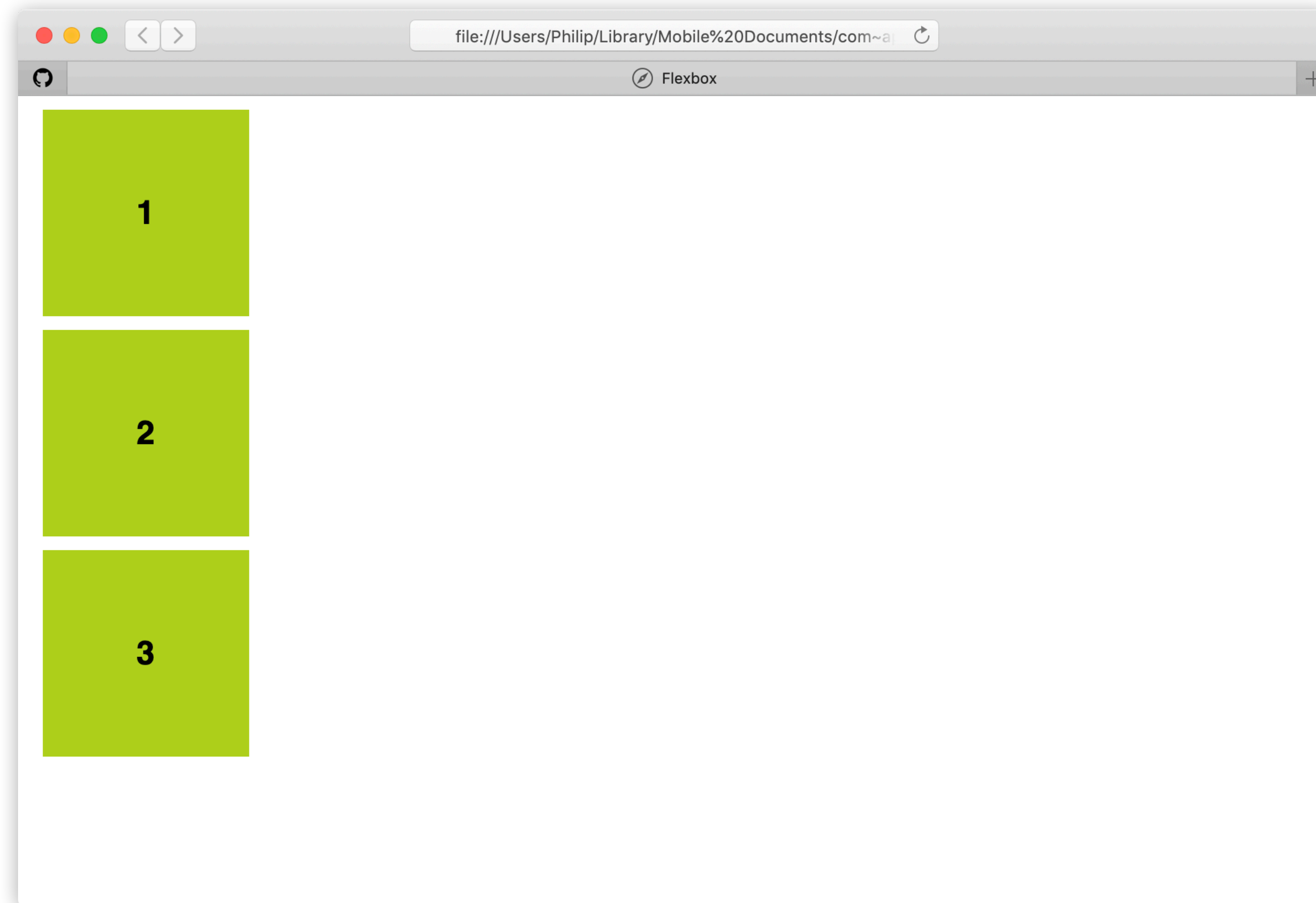
HTML/CSS: SETUP

```
<div class="flex">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
</div>
```

```
.box {
  width: 150px;
  height: 150px;
  margin: 10px;
  background-color: #acd123;
}
```

CODE

HTML/CSS: SETUP

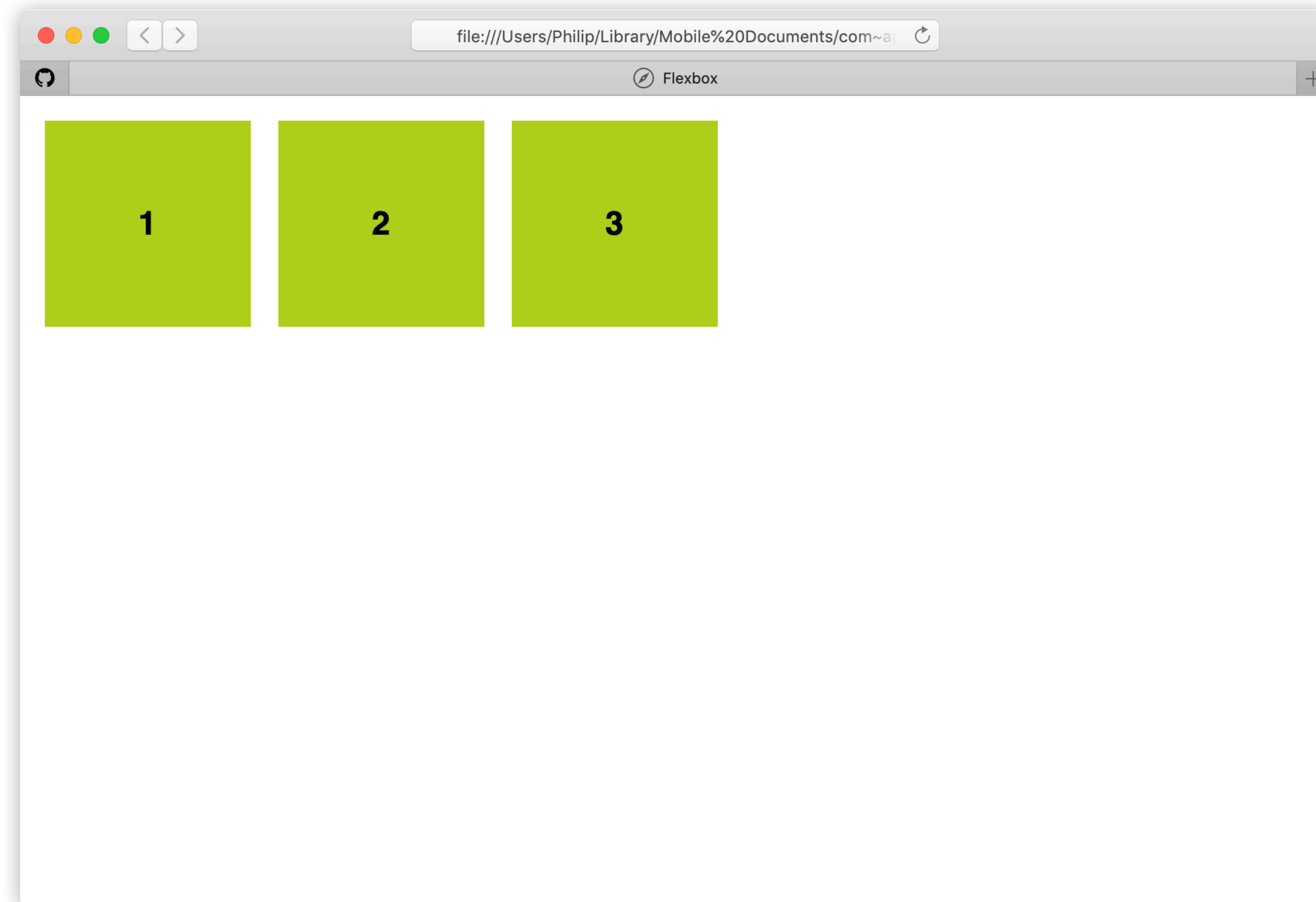


FLEXBOX

```
.flex {  
  display: flex;  
}
```

CODE

FLEXBOX



FLEX-DIRECTION

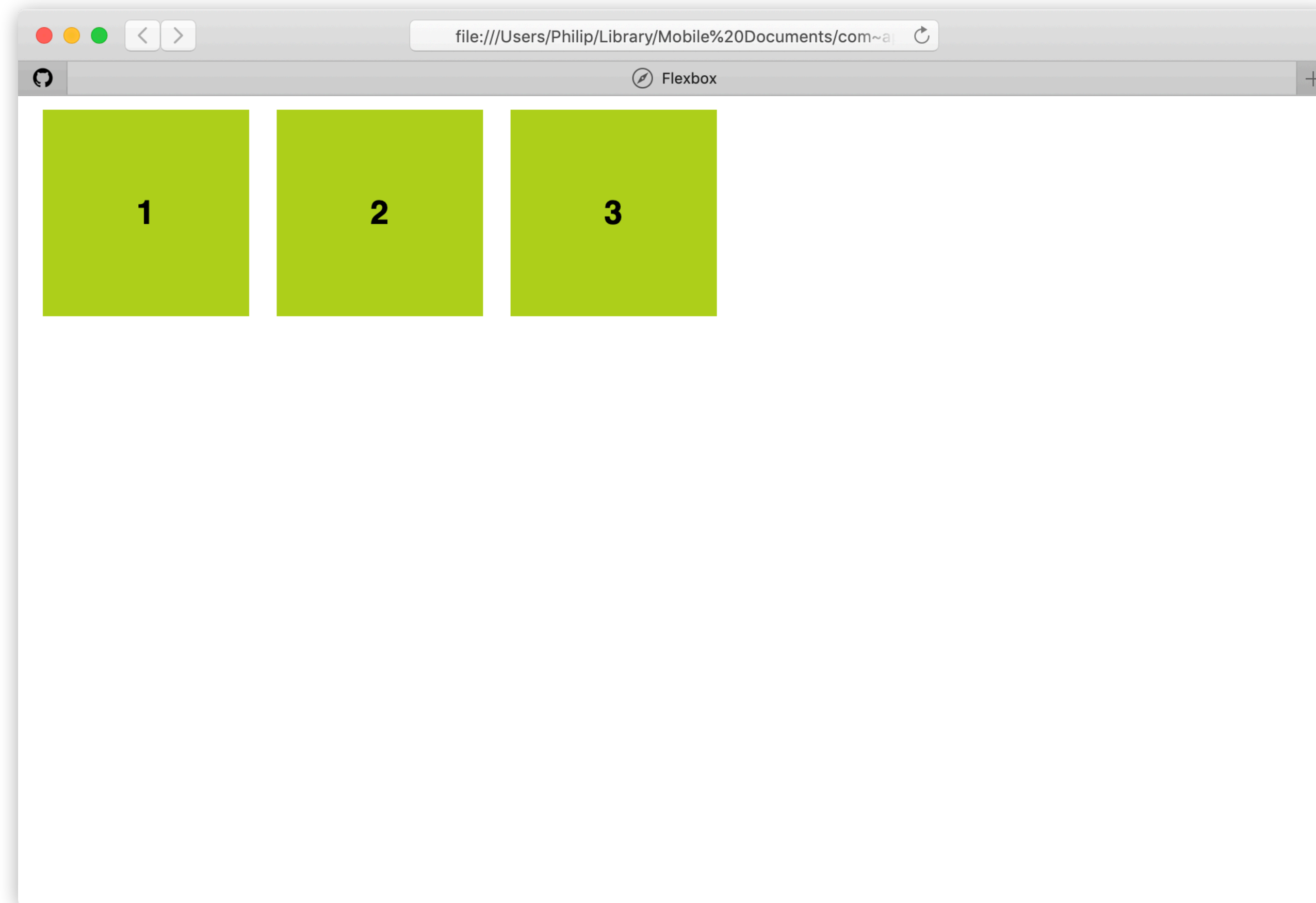
FLEXBOX: FLEX-DIRECTION

```
.flex {  
  display: flex;  
  flex-direction: row; /* Establishes main-axis along horizontal. Default is row */  
}
```

CODE



FLEXBOX: FLEX-DIRECTION



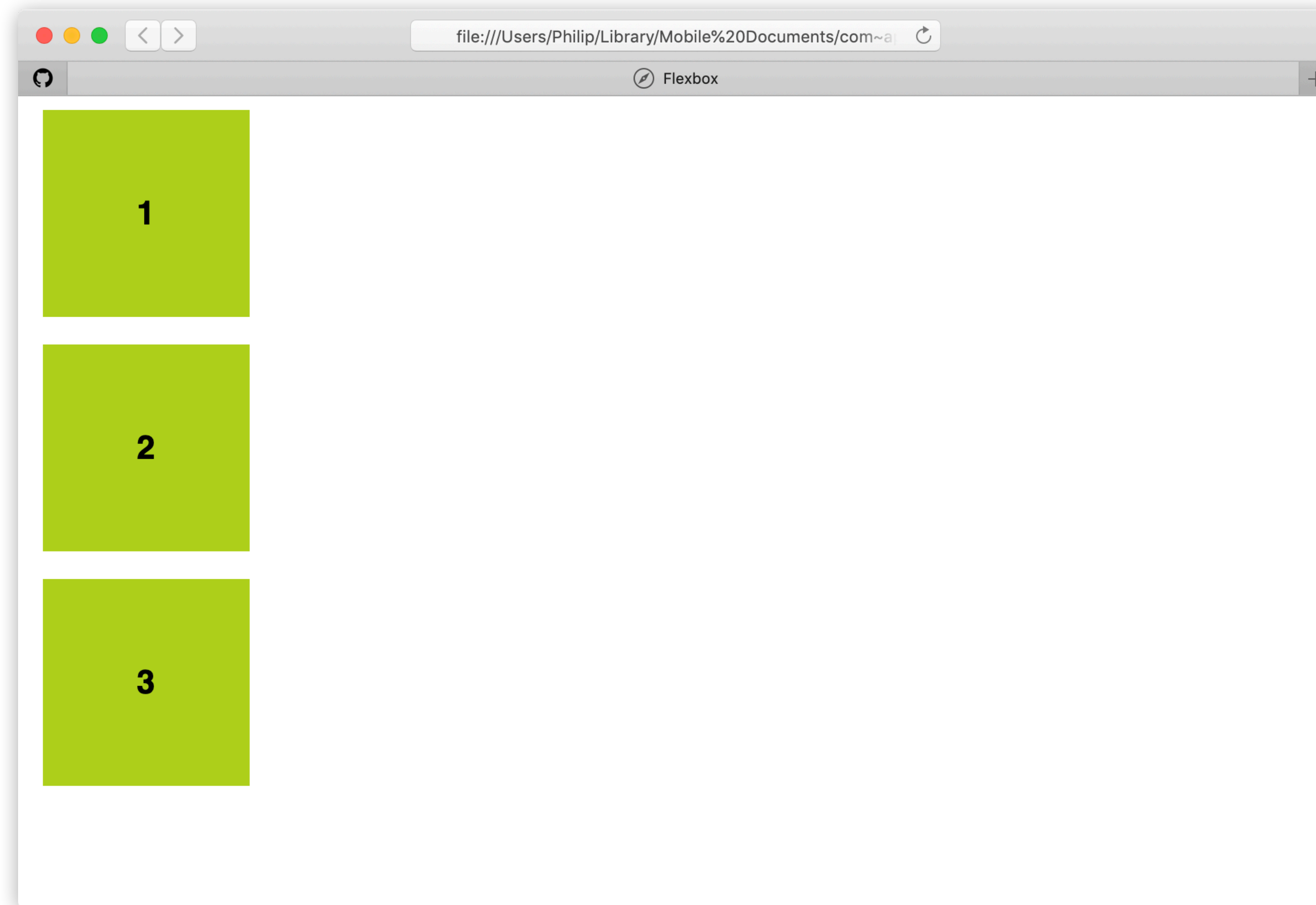
FLEXBOX: FLEX-DIRECTION

```
.flex {  
  display: flex;  
  flex-direction: column; /* Establishes main-axis along vertical. */  
}
```

CODE



FLEXBOX: FLEX-DIRECTION



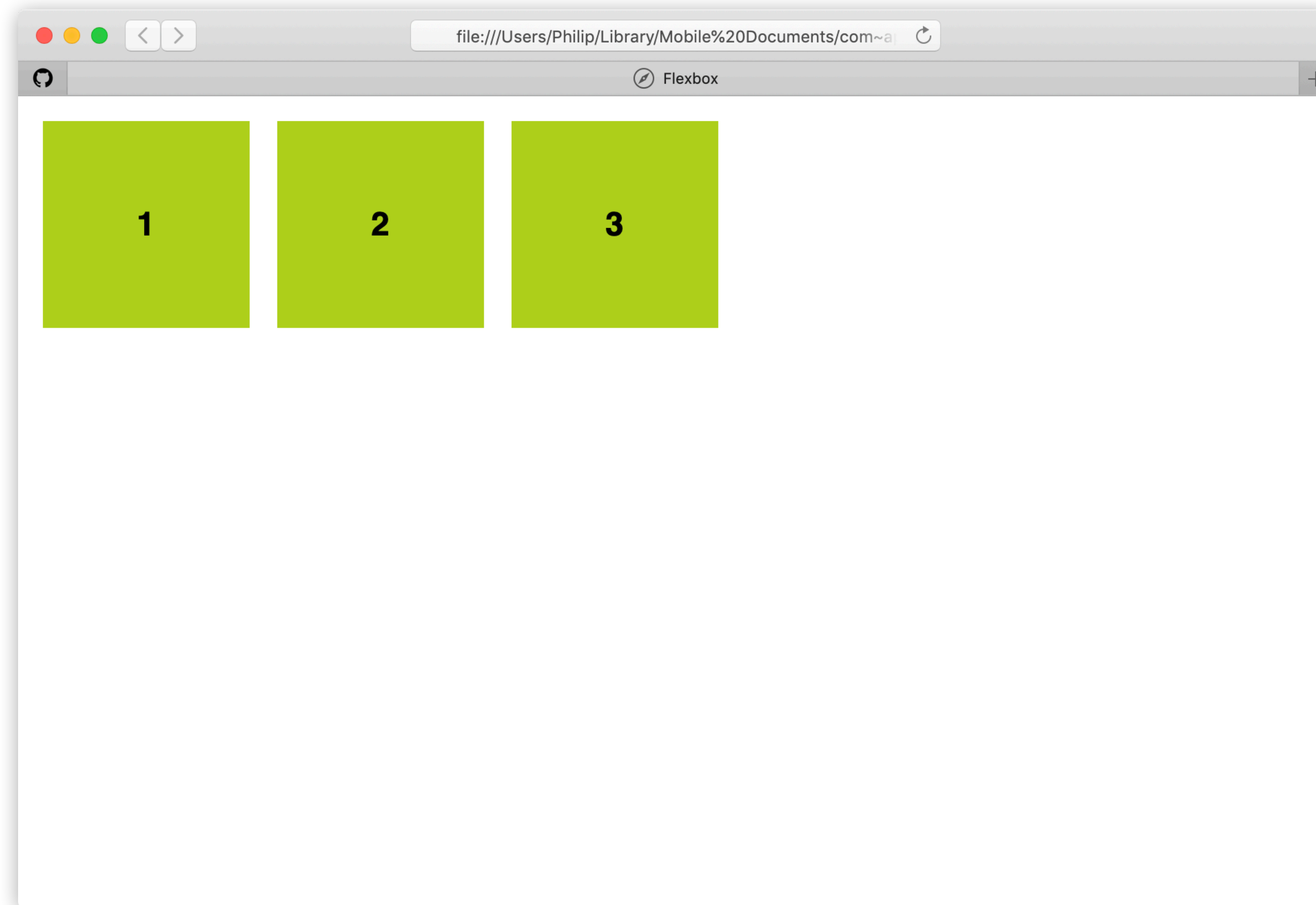
JUSTIFY-CONTENT

FLEXBOX: JUSTIFY-CONTENT

```
.flex {  
  display: flex;  
  justify-content: flex-start;  
}
```

CODE

FLEXBOX: JUSTIFY-CONTENT

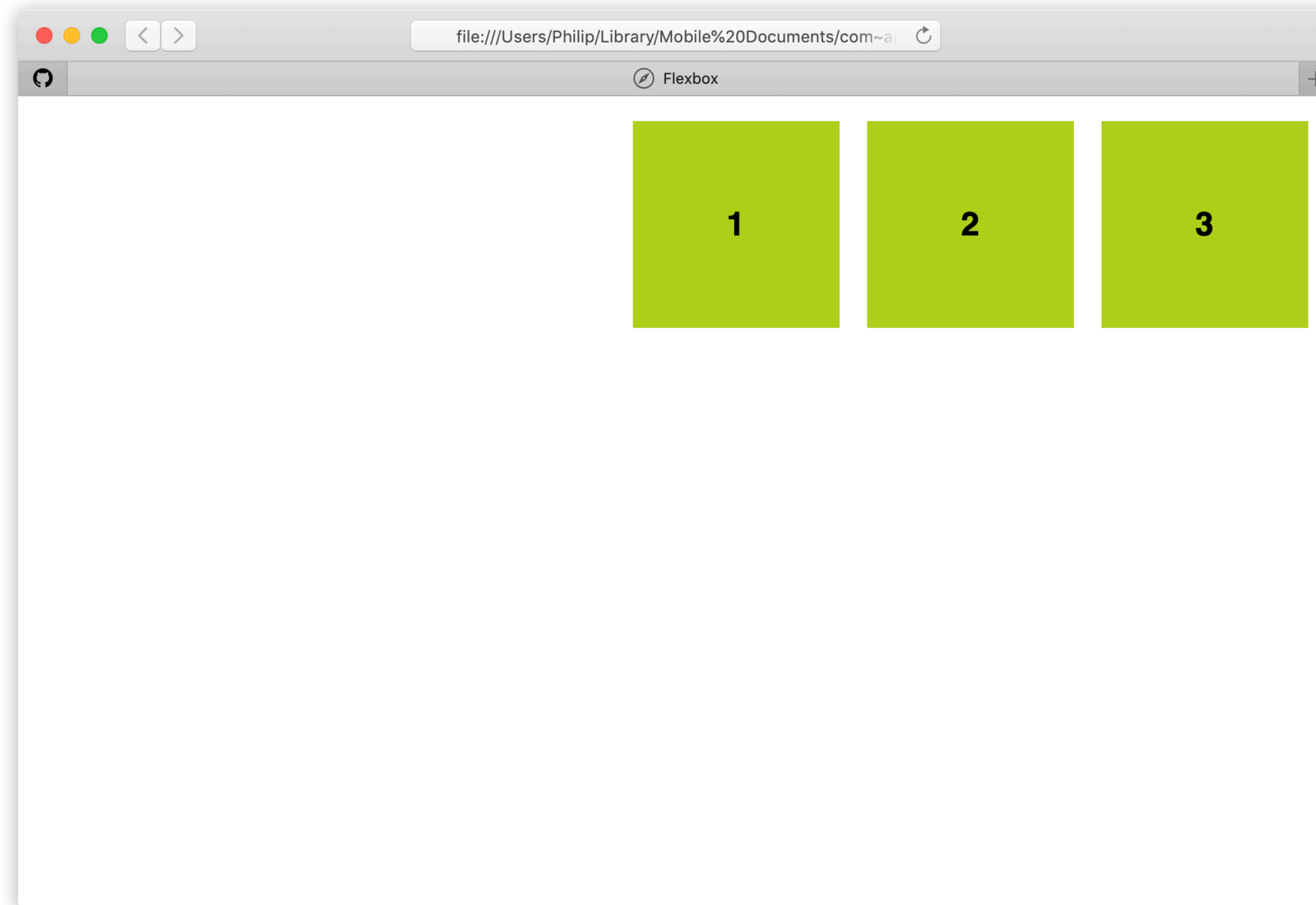


FLEXBOX: JUSTIFY-CONTENT

```
.flex {  
  display: flex;  
  justify-content: flex-end;  
}
```

CODE

FLEXBOX: JUSTIFY-CONTENT



FLEX-START & FLEX-END

`flex-start` and `flex-end` are dependent on `flex-direction` e.g.:

FLEX-START & FLEX-END

`flex-start` and `flex-end` are dependent on `flex-direction` e.g.:

```
.flex {  
  display: flex;  
  flex-direction: row;          /* Flex items on horizontal axis */  
  justify-content: flex-start; /* start is left. end is right. */  
}
```


FLEX-START & FLEX-END

`flex-start` and `flex-end` are dependent on `flex-direction` e.g.:

```
.flex {
  display: flex;
  flex-direction: row;          /* Flex items on horizontal axis */
  justify-content: flex-start; /* start is left. end is right. */
}
```

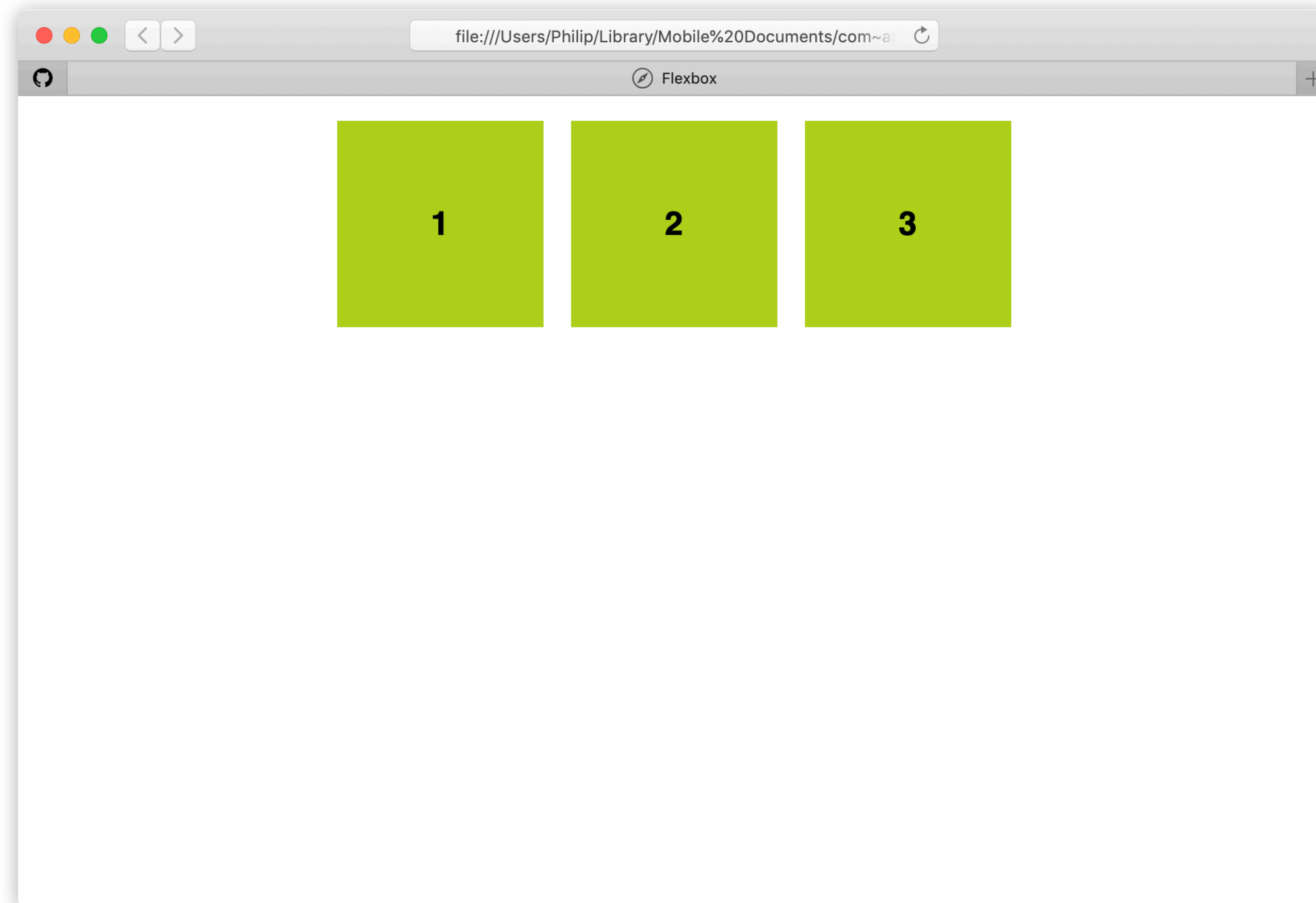
```
.flex {
  display: flex;
  flex-direction: column;      /* Flex items on vertical axis */
  justify-content: flex-start; /* start is top. end is bottom. */
}
```

FLEXBOX: JUSTIFY-CONTENT

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

CODE

FLEXBOX: JUSTIFY-CONTENT

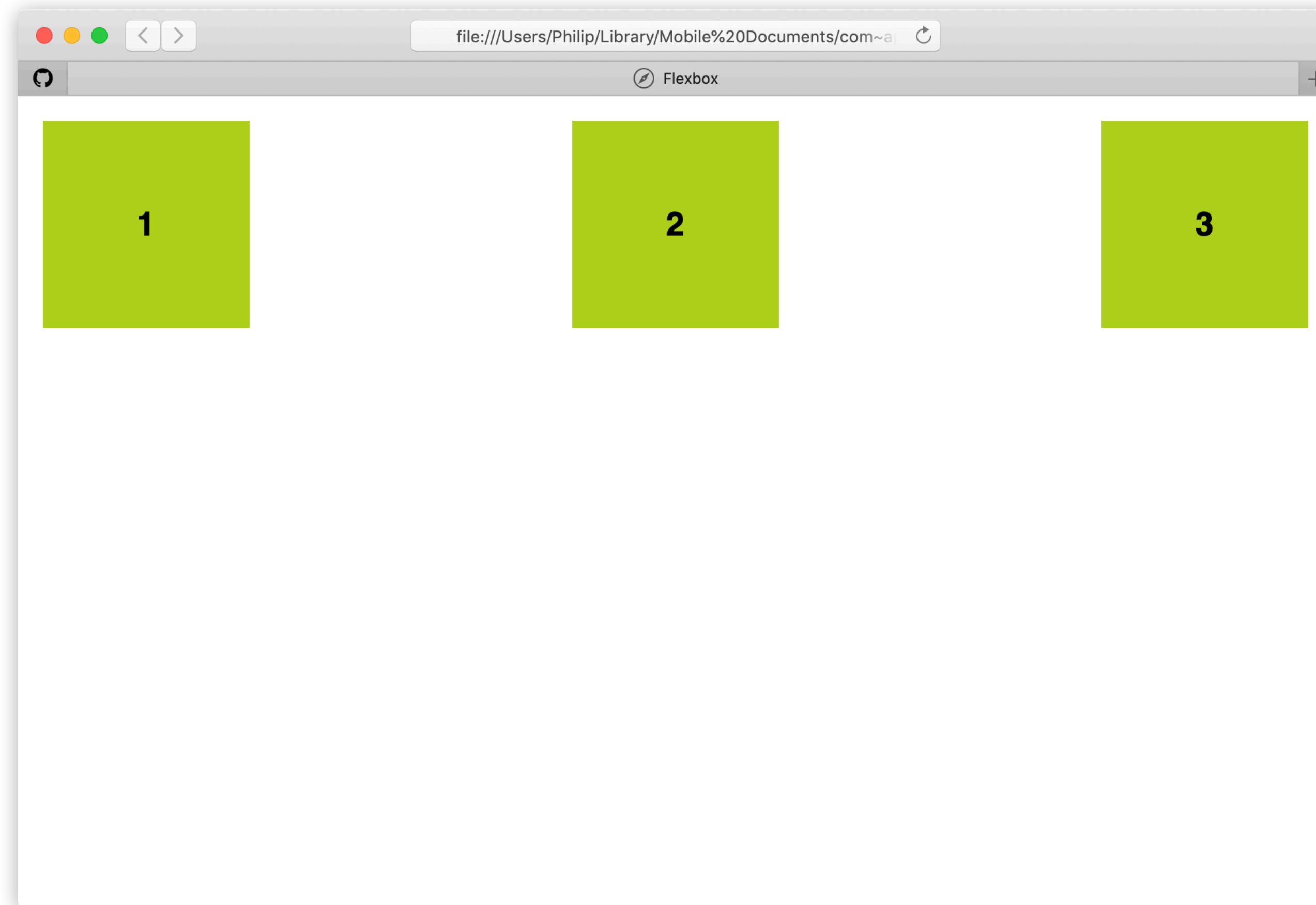


FLEXBOX: JUSTIFY-CONTENT

```
.flex {  
  display: flex;  
  justify-content: space-between;  
}
```

CODE

FLEXBOX: JUSTIFY-CONTENT

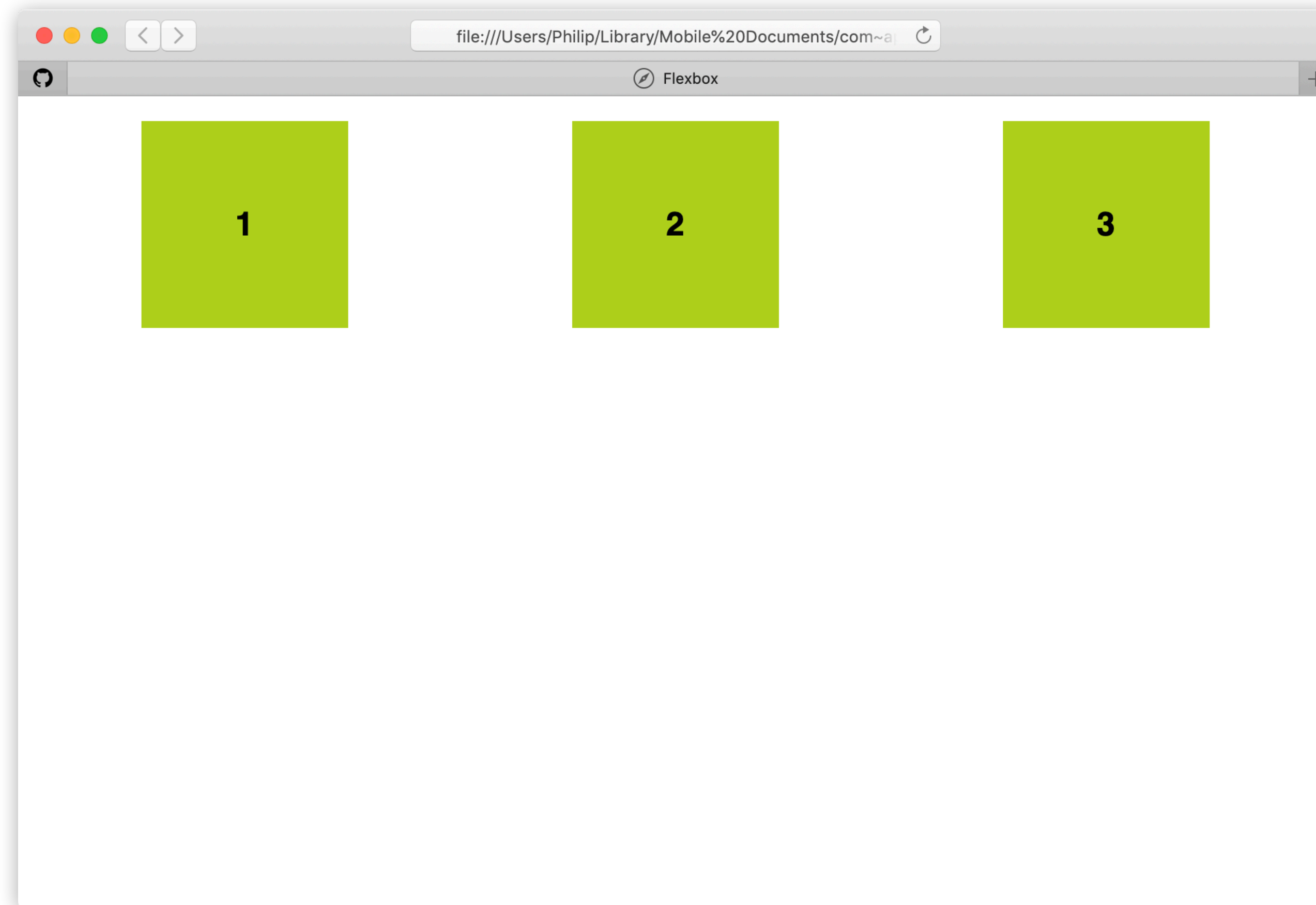


FLEXBOX: JUSTIFY-CONTENT

```
.flex {  
  display: flex;  
  justify-content: space-around;  
}
```

CODE

FLEXBOX: JUSTIFY-CONTENT



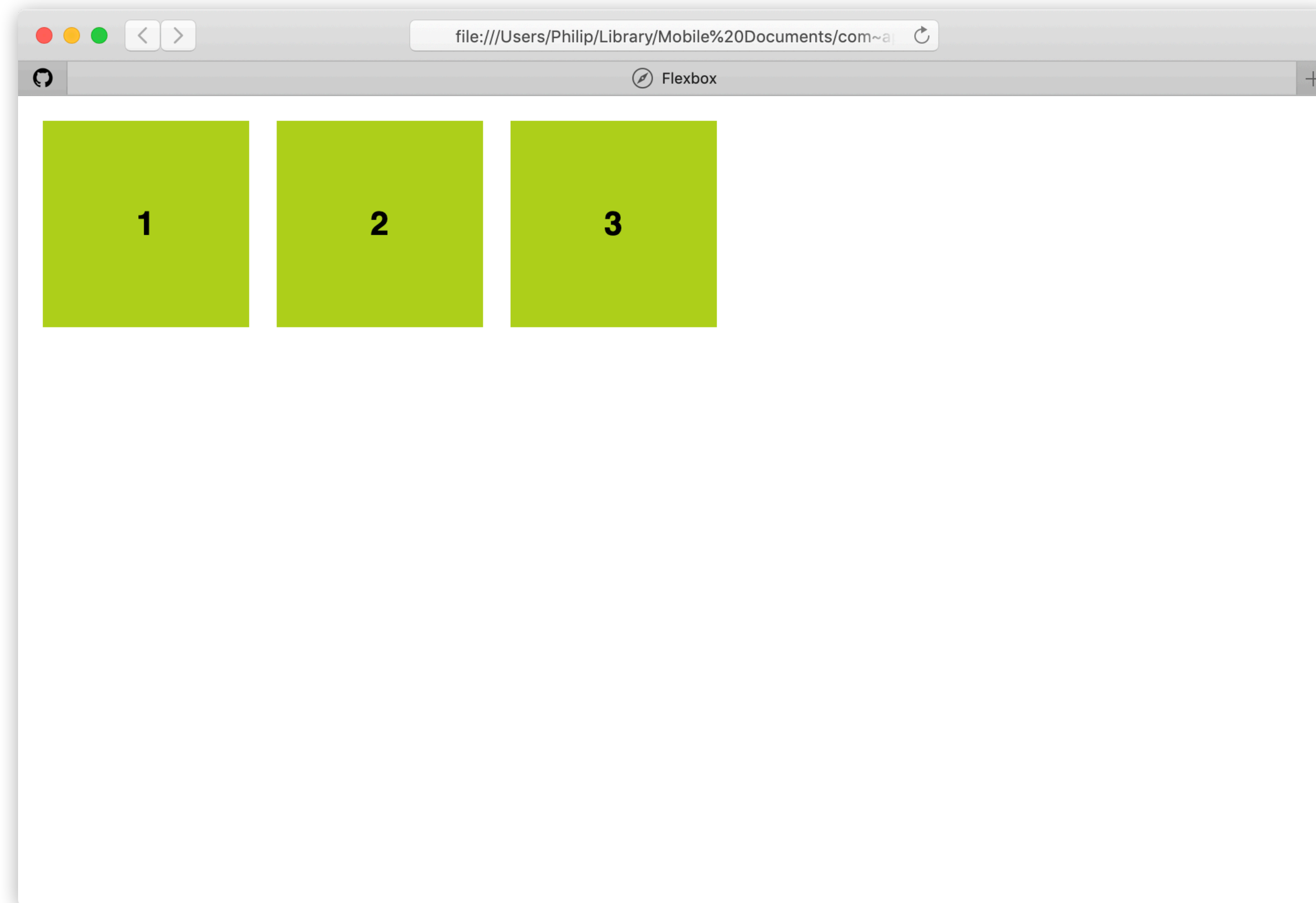
ALIGN-ITEMS

FLEXBOX: ALIGN-ITEMS

```
.flex {  
  display: flex;  
  align-items: flex-start; /* Similar to justify-content, but opposite axis */  
}
```

CODE

FLEXBOX: ALIGN-ITEMS

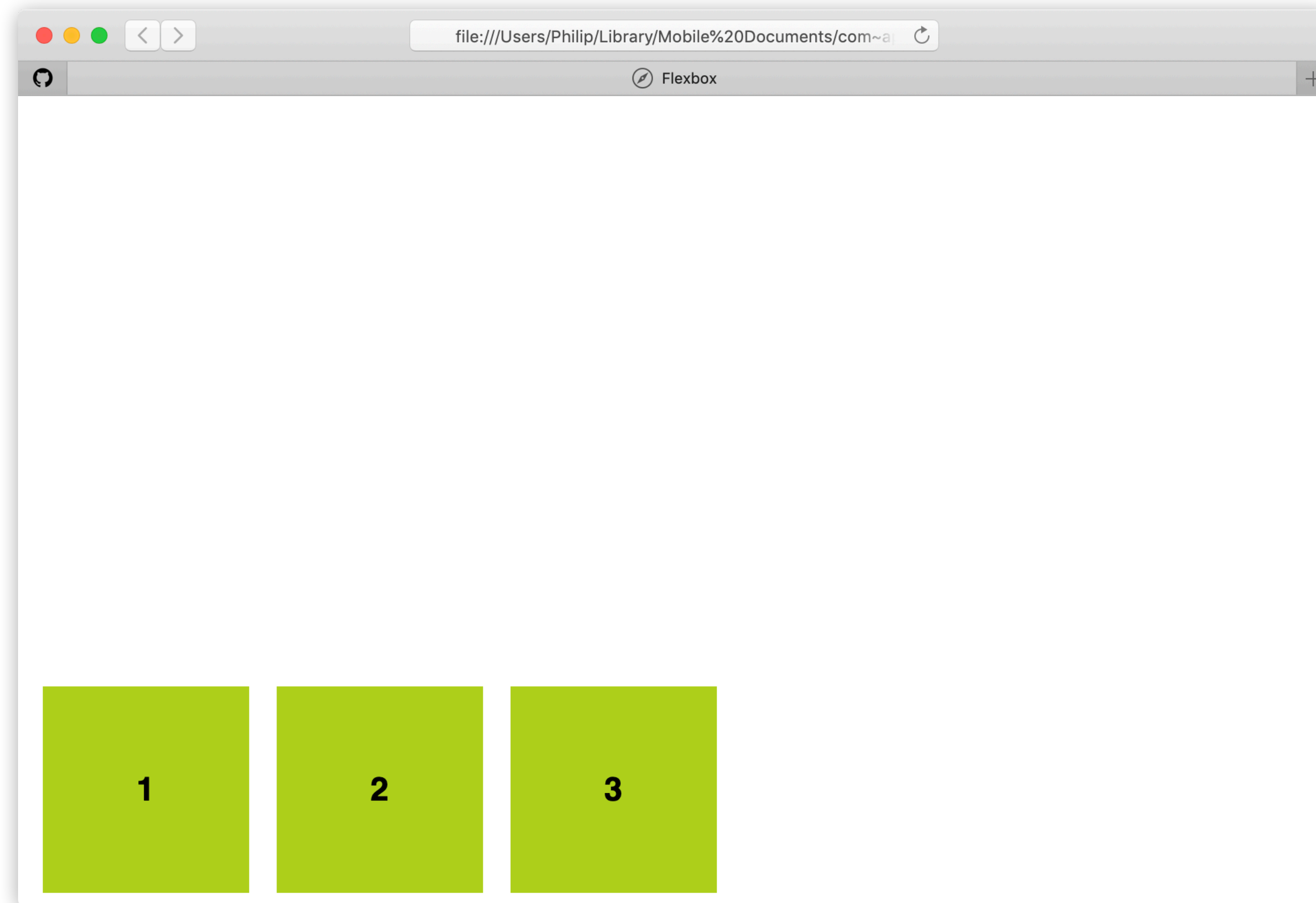


FLEXBOX: ALIGN-ITEMS

```
.flex {  
  display: flex;  
  align-items: flex-end;  
}
```

CODE

FLEXBOX: ALIGN-ITEMS

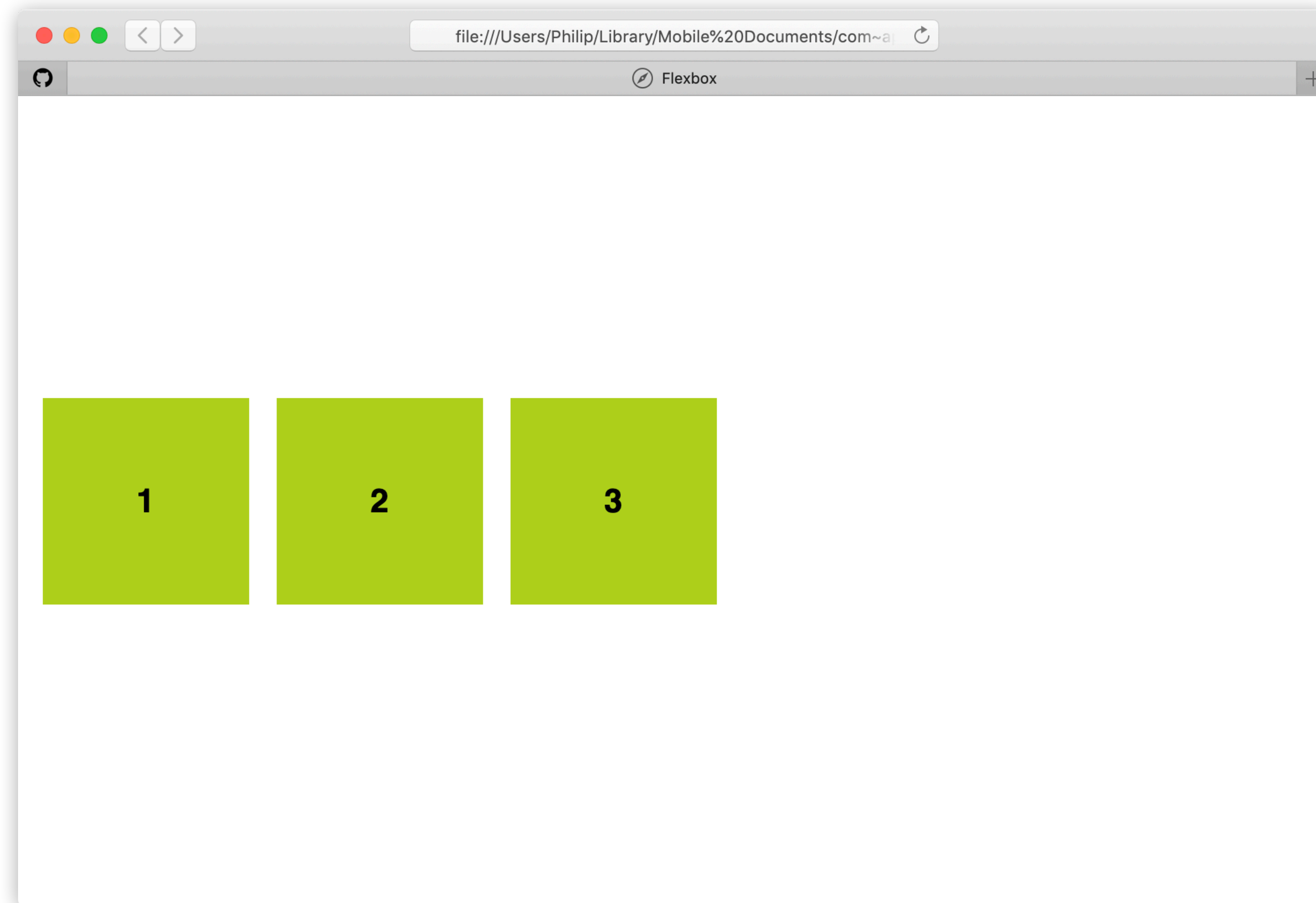


FLEXBOX: ALIGN-ITEMS

```
.flex {  
  display: flex;  
  align-items: center;  
}
```

CODE

FLEXBOX: ALIGN-ITEMS

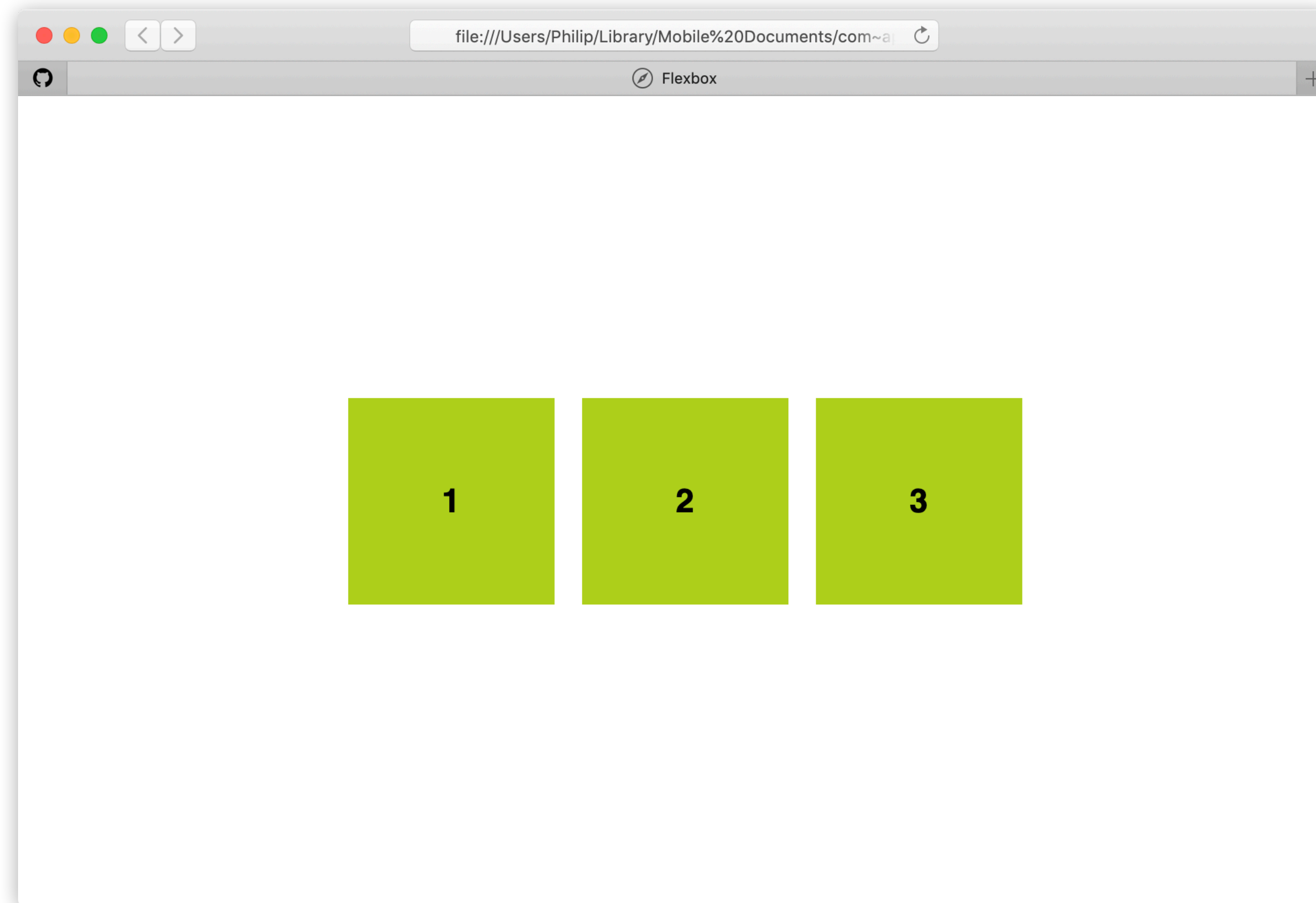


FLEXBOX: CENTER

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

CODE

FLEXBOX: CENTER



<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

QUESTIONS?