



Make **CSS** great again

"Una Historia de Layouts"

Pedro A. Sonzini Astudillo



/tretapey

Repo: **webconf-talk**

Había una vez...

Cuenta la leyenda...

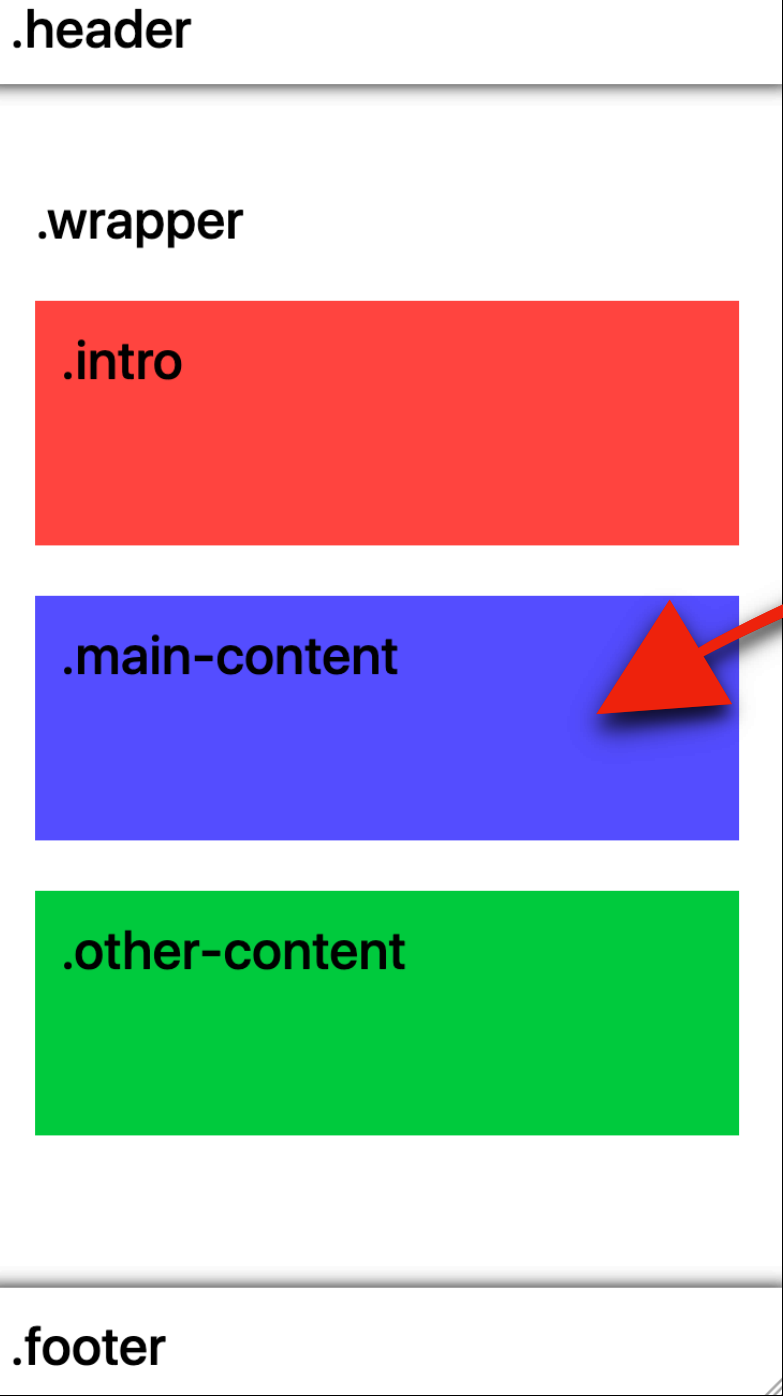
Hace mucho tiempo en
una galaxia muy muy
lejana...

Tablas

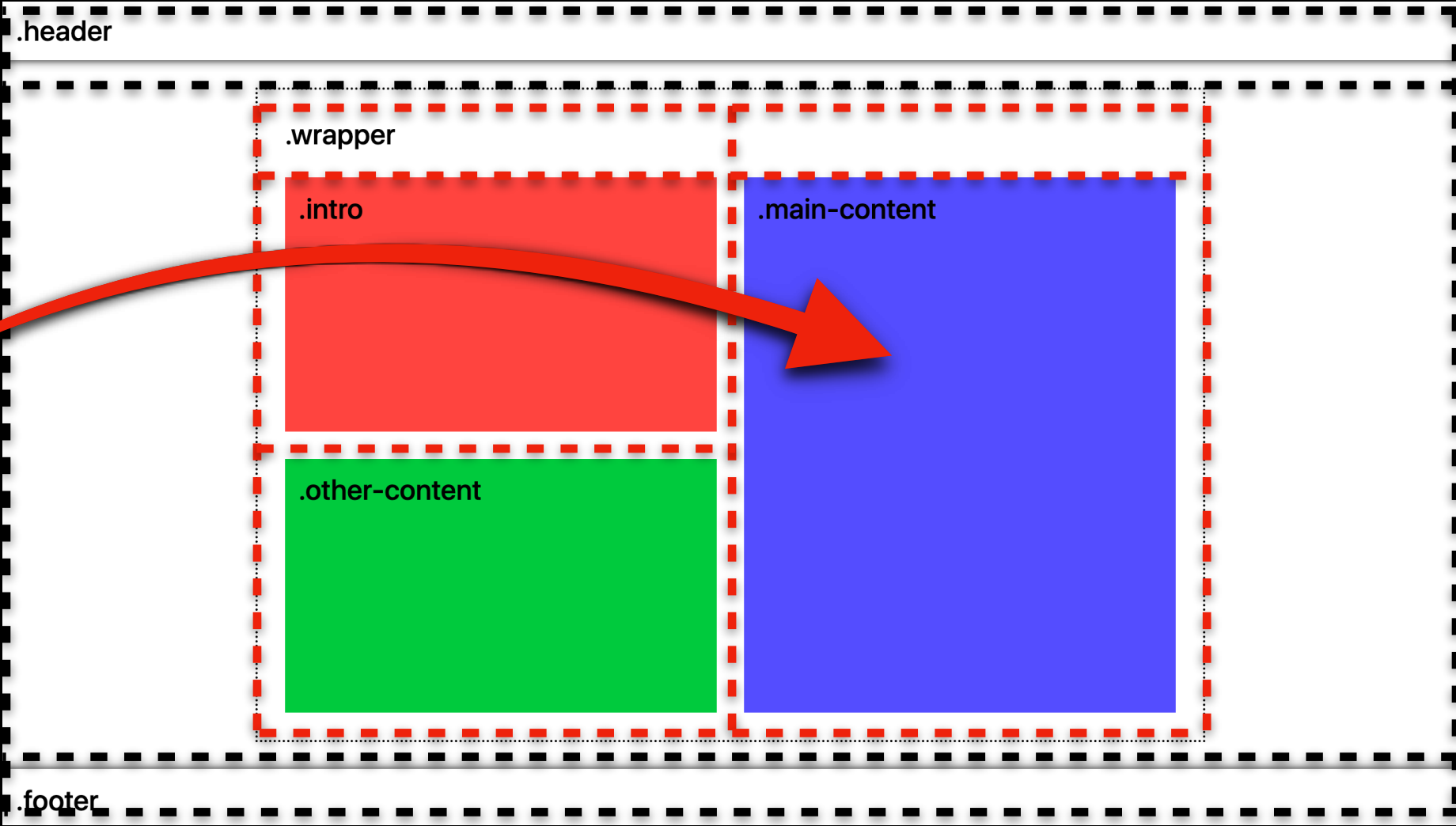
Algunas características para rescatar...

- **Son Bi-Dimensionales**
- **Se pueden anidar**
- **Son "dinámicas"... en cierto modo**
- **Permiten espacios en blanco**

Mobile

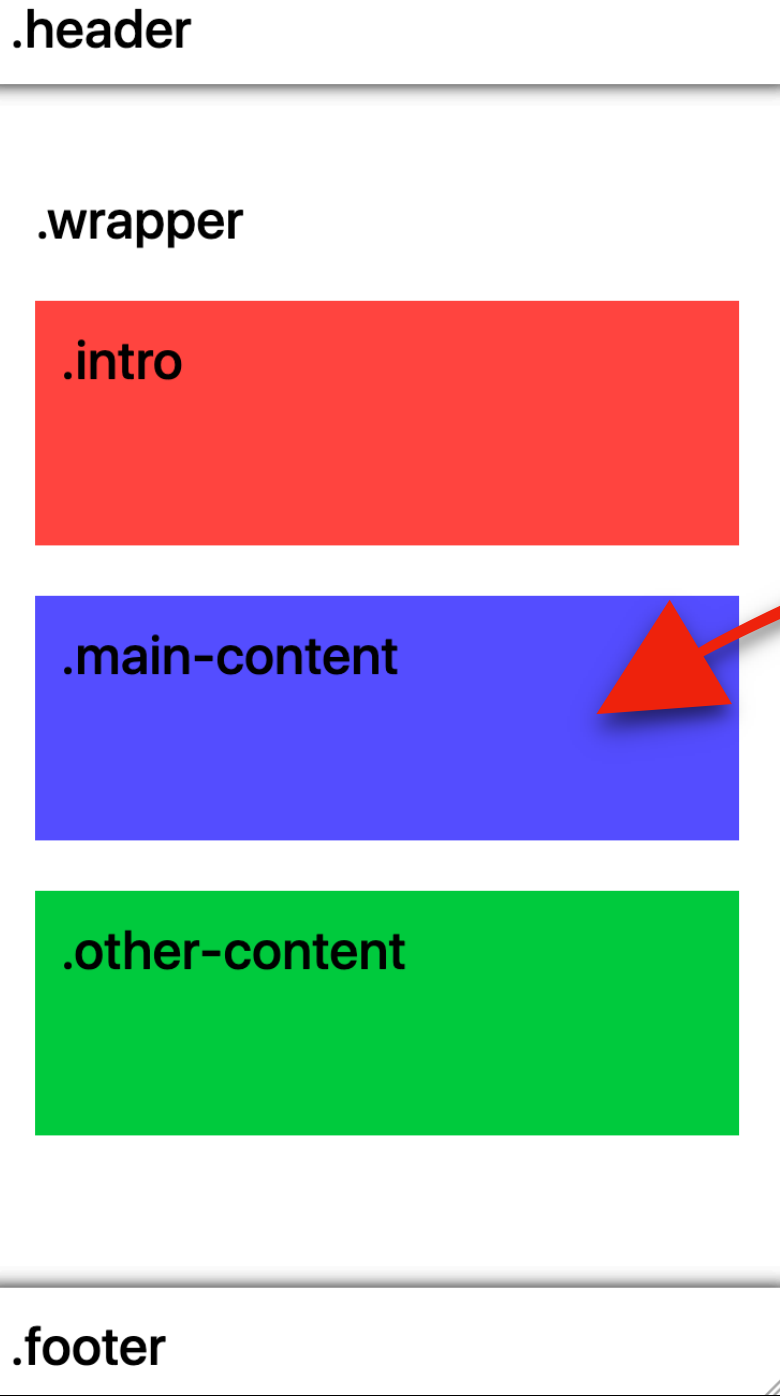


Desktop

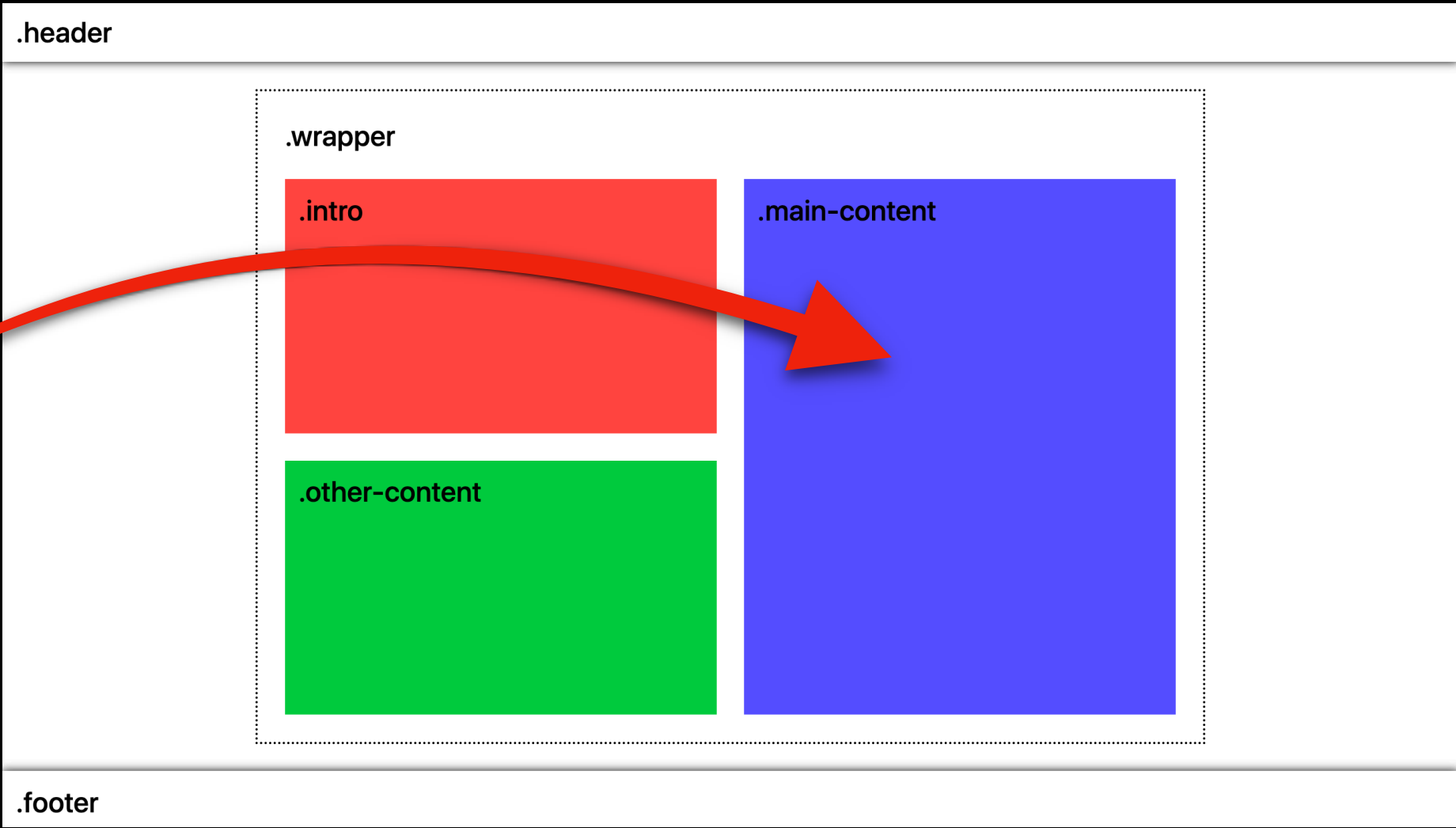


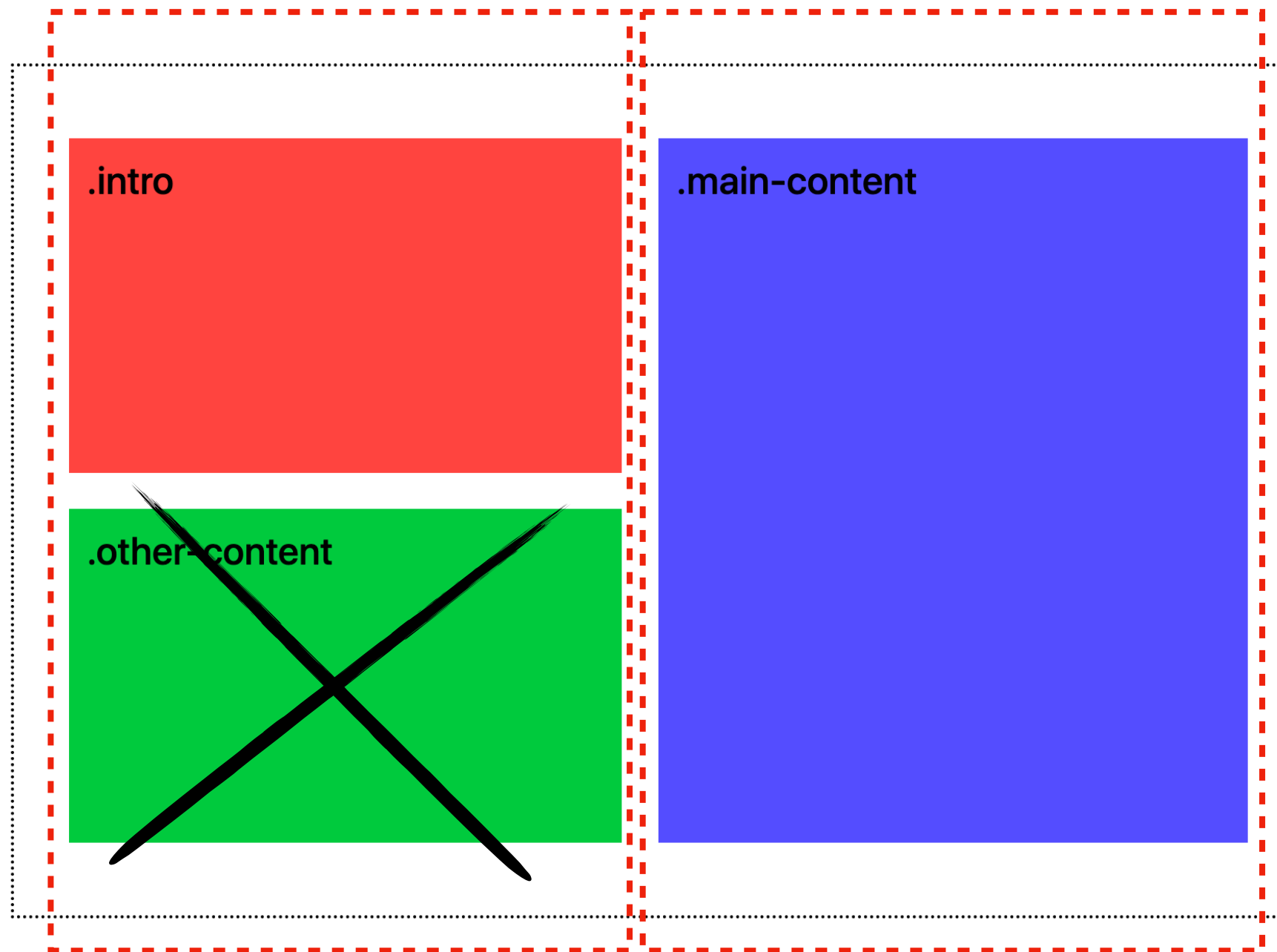
**No usen tablas para
hacer layouts!!!**

Mobile



Desktop





.intro

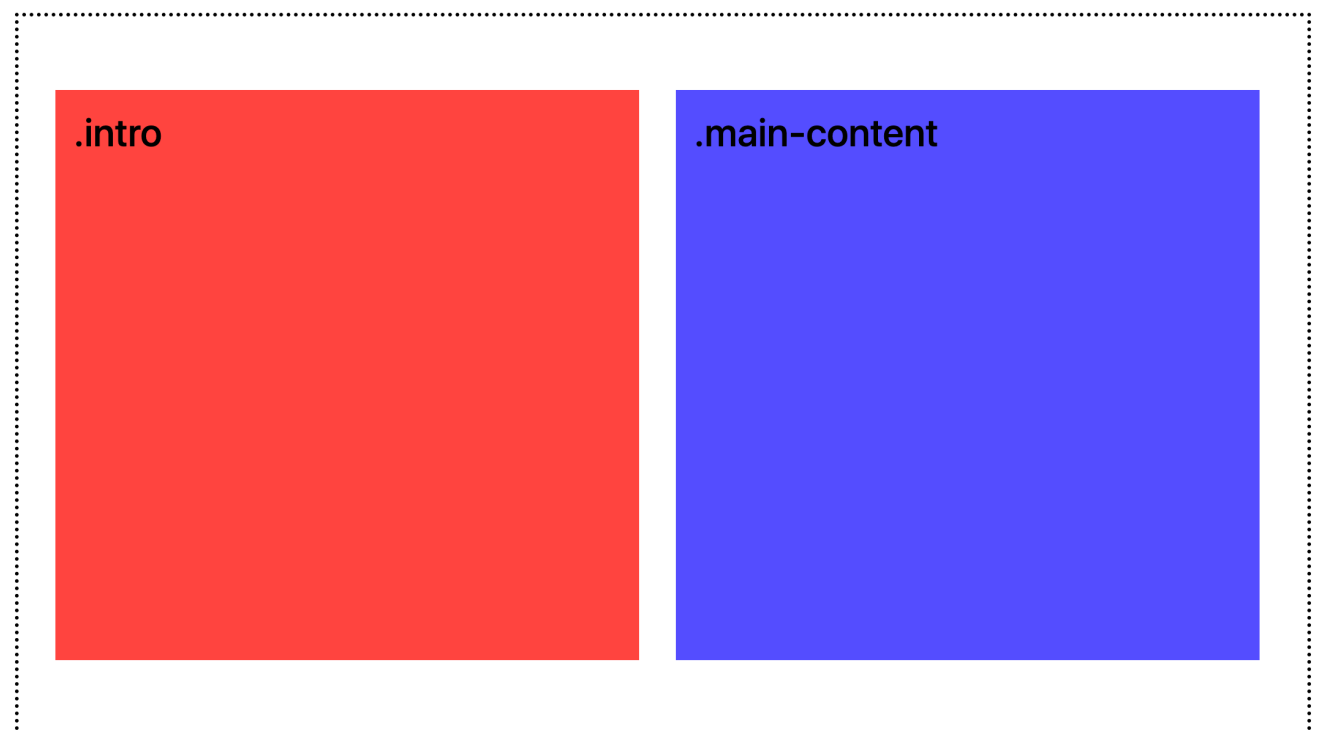
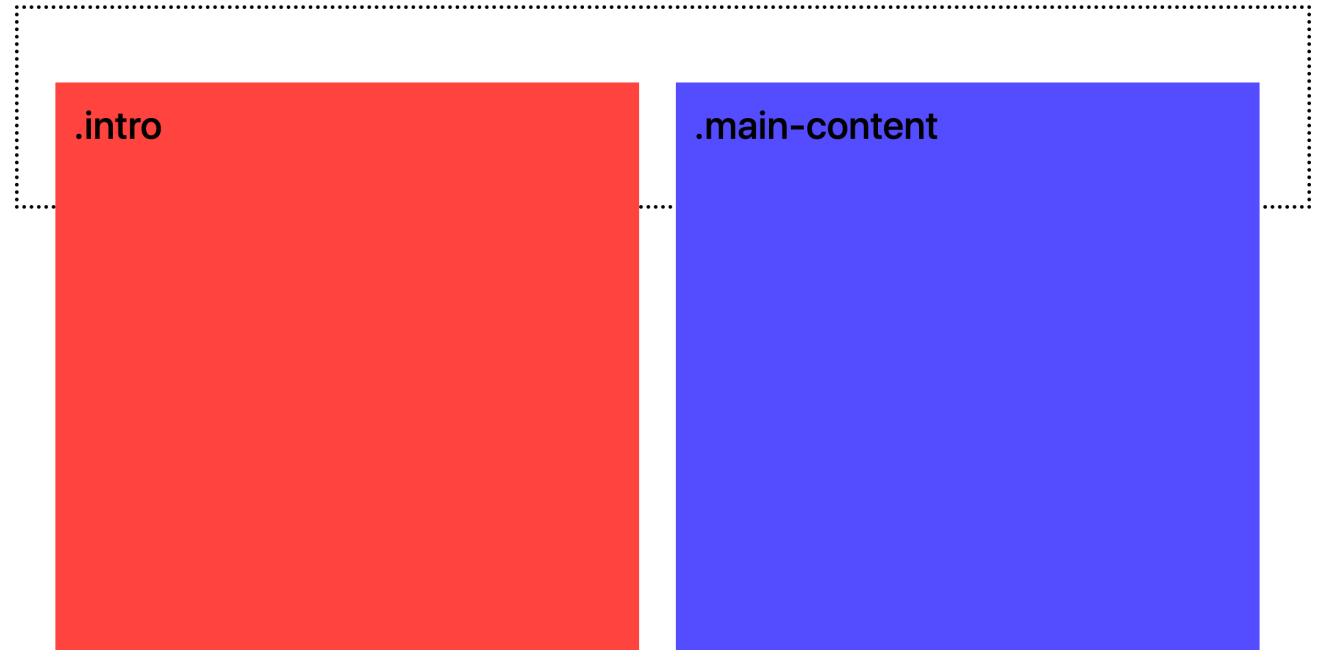
.main-content

Float

```
.wrapper {  
  position: relative;  
  padding: 30px;  
  margin: 30px auto;  
}
```

```
.intro {  
  float: left;  
}
```

```
.main-content {  
  float: left;  
}
```

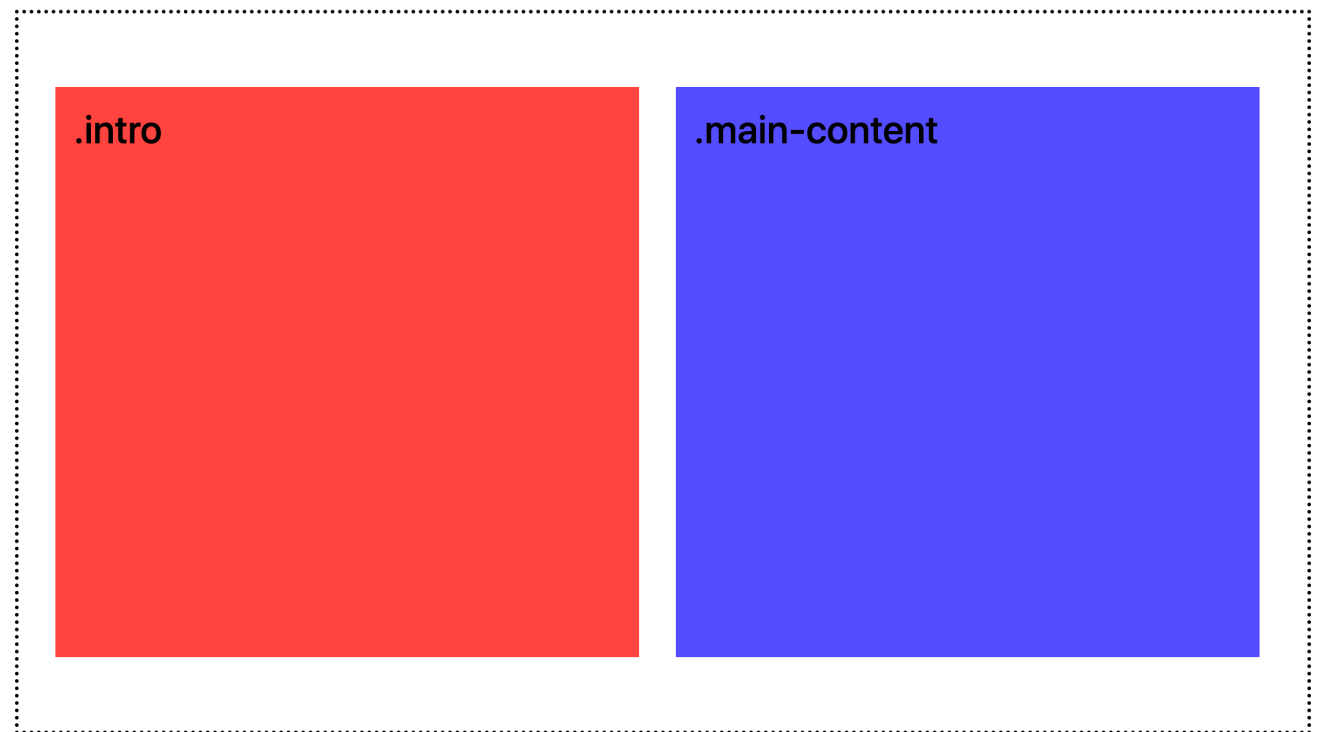


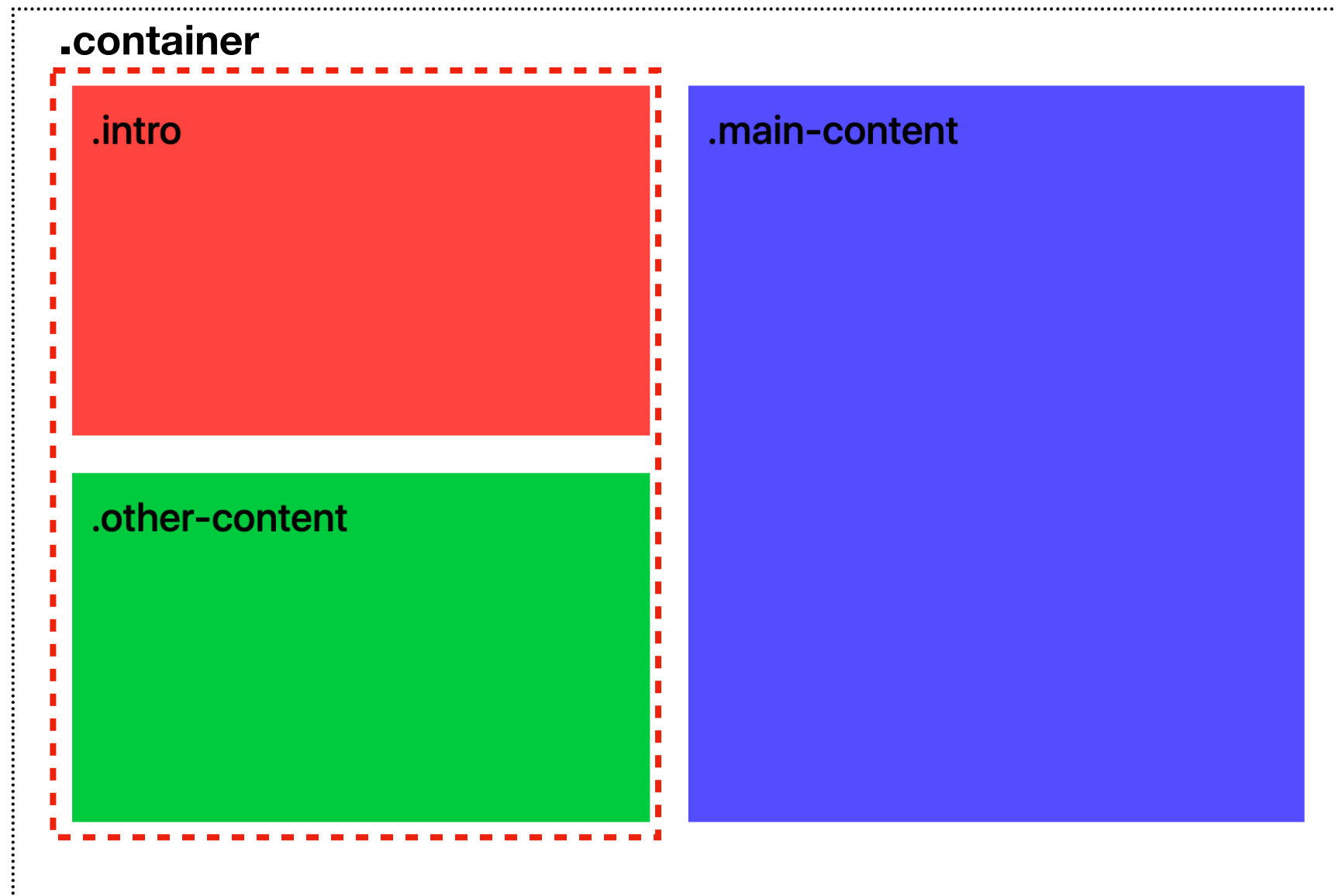
Flexbox

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

```
.intro {  
  width: 45%;  
  background: ■ #FF4848;  
}
```

```
.main-content {  
  width: 45%;  
  background: ■ #5757FF;  
}
```





```

<div class="wrapper">

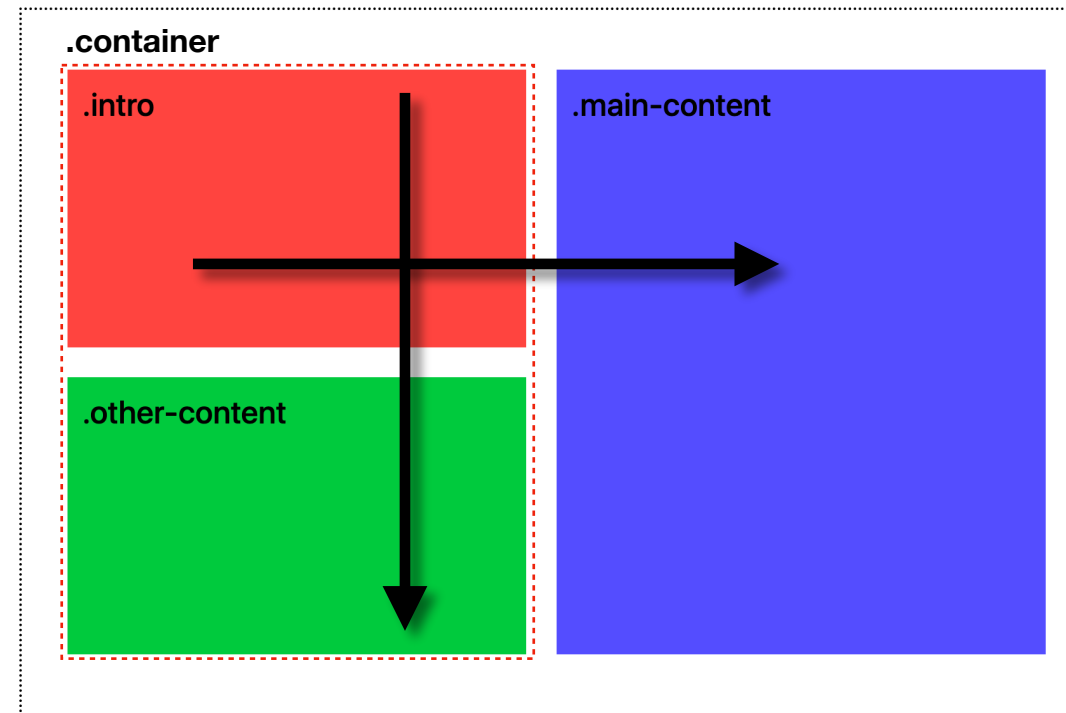
  <div class="container">
    <div class="intro">
      <!-- Some content -->
    </div>

    <div class="other-content">
      <!-- Some content -->
    </div>
  </div>

  <main class="main-content">
    <!-- Some content -->
  </main>
</div>

```

.wrapper



```

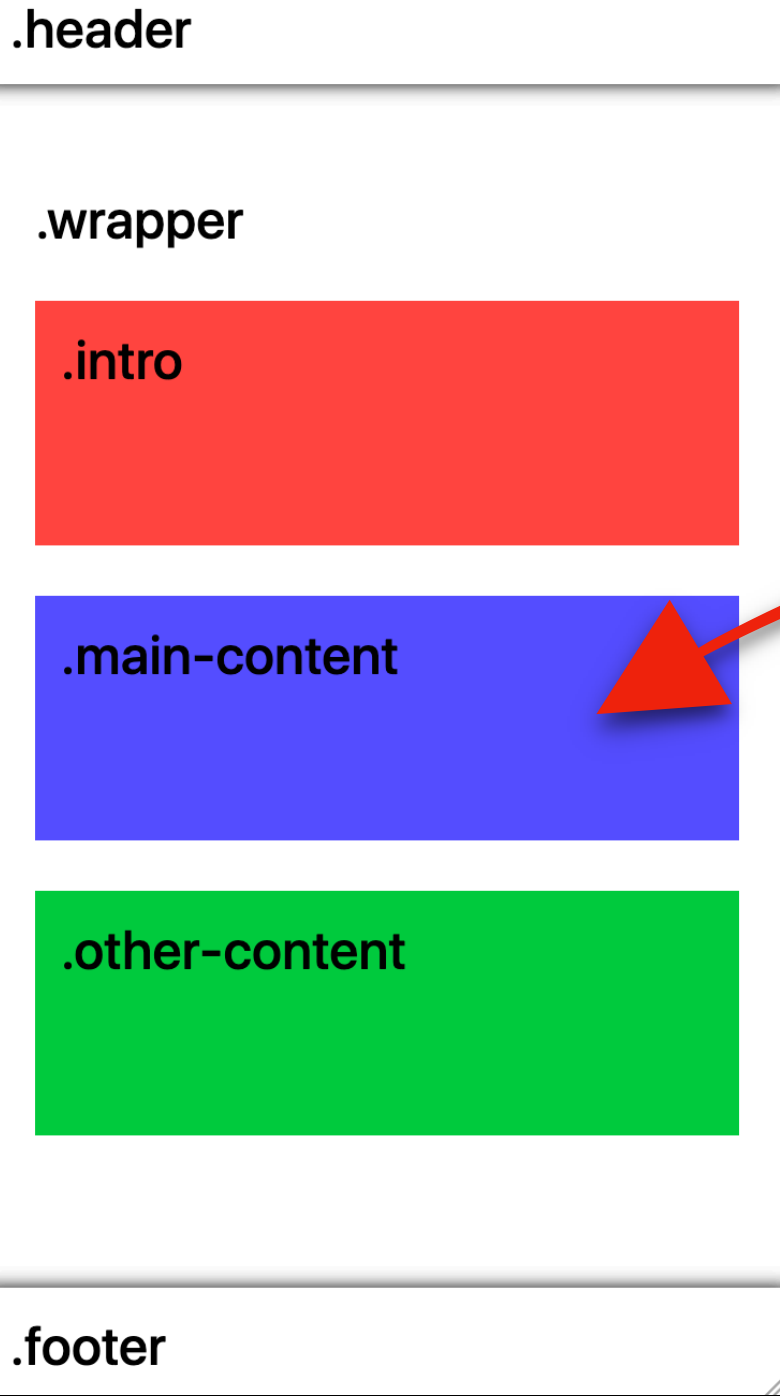
.wrapper {
  display: flex;
  justify-content: space-between;
}

.container {
  border: 1px dotted red;
  display: flex;
  flex-direction: column;
  align-items: space-between;
}

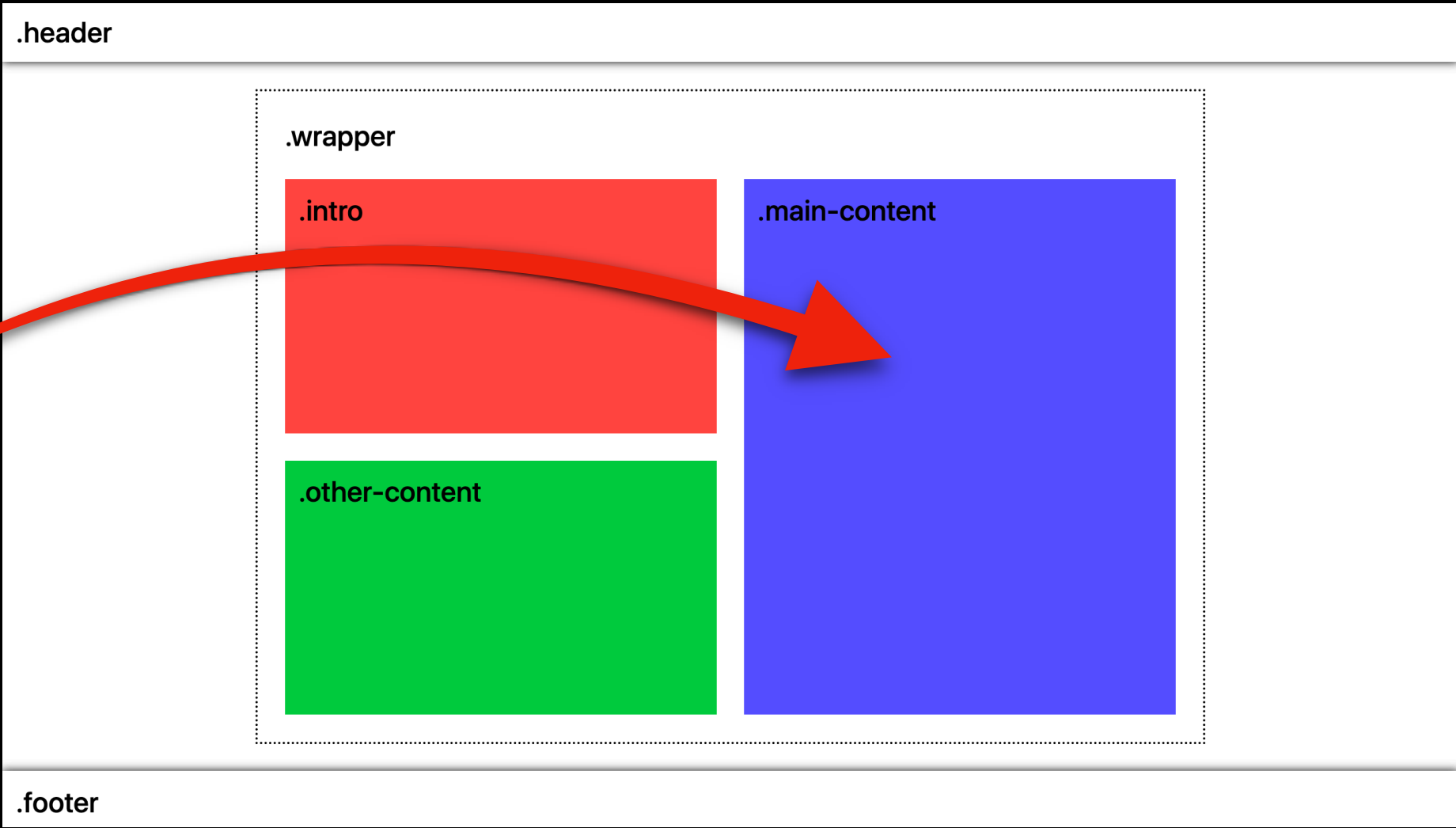
.main-content, .intro, .other-content {
  flex-basis: 0;
  flex-grow: 1;
}

```

Mobile



Desktop



```
<div class="wrapper">
  <!-- <h1>.wrapper</h1> -->

  <div class="container">
    <div class="intro">
      <h1>.intro</h1>
    </div>

    <div class="other-content">
      <h1>.other-content</h1>
    </div>
  </div>

  <main class="main-content">
    <h1>.main-content</h1>
  </main>

  <div class="other-content-mobile">
    <h1>.other-content</h1>
  </div>

</div>
```

```
.other-content {
  display: none;
}

.other-content-mobile {
  display: block;
}

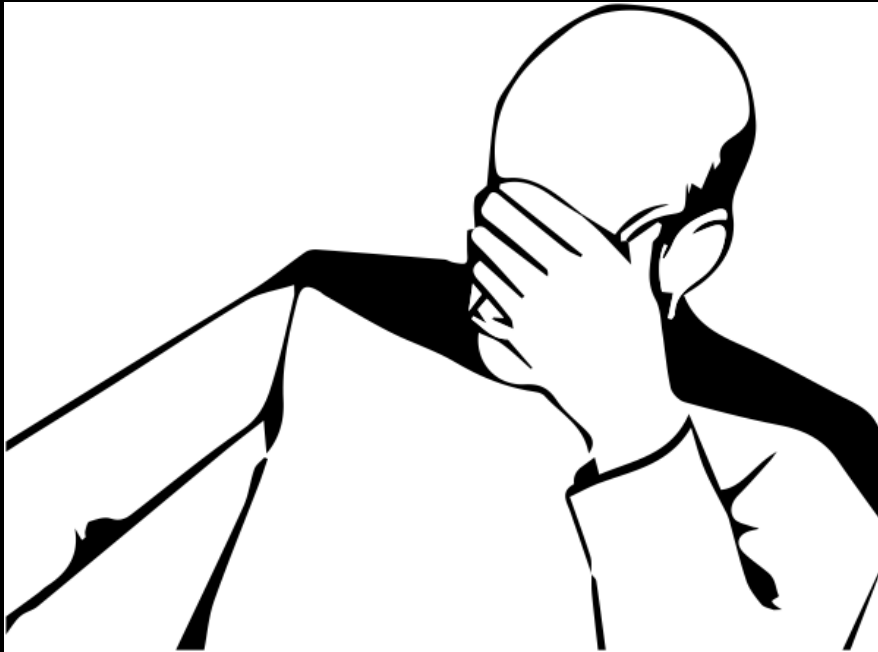
@media(min-width:992px) {
  .wrapper {
    width: 992px;
    display: flex;
    justify-content: space-between;
  }

  .container {
    display: flex;
    flex-direction: column;
  }

  .other-content {
    display: block;
  }

  .other-content-mobile {
    display: none;
  }
}
```


Bootstrap



```
<!-- Wrapper -->
<div class="container">
  <div class="row">

    <!-- Container -->
    <div class="col-lg-6">
      <div class="row">

        <div class="col-12">
          <!-- Intro -->
        </div>

        <div class="col-12 d-none d-lg-block">
          <!-- Other content desktop -->
        </div>

      </div>
    </div>

    <div class="col-lg-6">
      <!-- Main content -->
    </div>

    <div class="col-12 d-lg-none">
      <!-- Other content mobile -->
    </div>
  </div>
</div>
```

La WEB, los layouts, están rotos...
Con el tiempo encontramos la
forma de ~~arreglarla~~ hacer trampa...
evolucionamos (?)

Mobile

.header

.wrapper

.intro

.main-content

.other-content

.footer

```
<body>
  <header class="header"></header>

  <div class="wrapper">
    <div class="intro"></div>

    <main class="main-content"></main>

    <div class="other-content"></div>
  </div>

  <footer class="footer"></footer>
</body>
```

La vista "mobile" nos ayuda a
definir el "markdown"
semántico... "Mobile first" ahora
tiene sentido, ¿no?

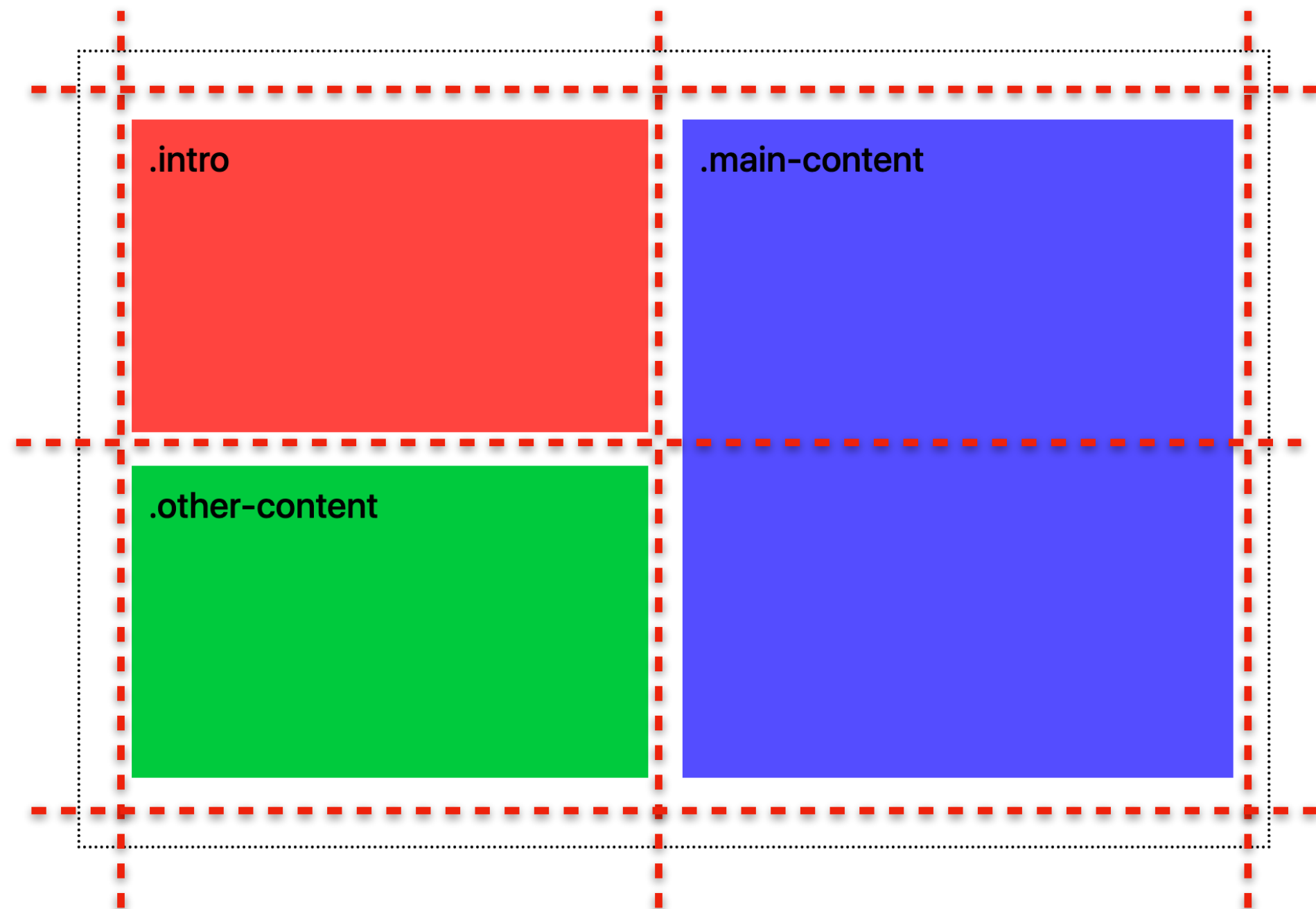
¡El futuro es hoy!

¡El futuro es hoy!

CSS grid

Algunas características para rescatar...

- **Bi-Dimensional**
- **Se pueden anidar**
- **Es dinámica**
- **Permite espacios en blanco**

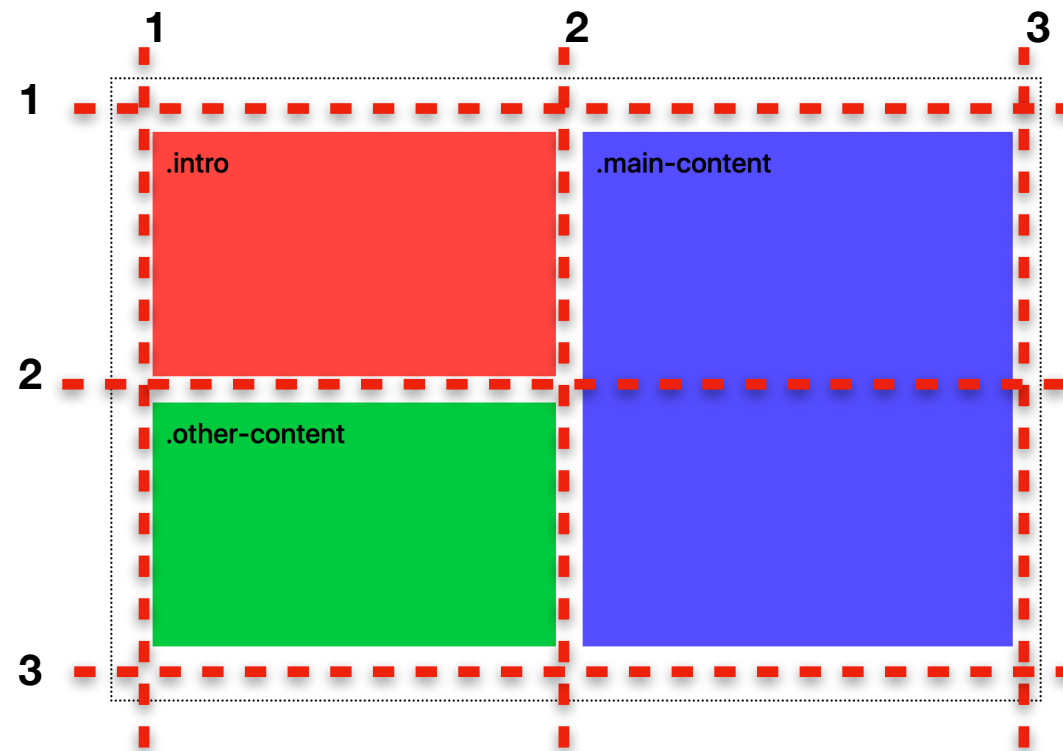


```

@media(min-width:992px) {
  .wrapper {
    width: 992px;
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: 1fr 1fr;
    grid-gap: 30px;
  }

  .main-content {
    grid-column: 2/3;
    grid-row: 1/3;
  }
}

```



```

<body>
  <header class="header"></header>

  <div class="wrapper">
    <div class="intro"></div>

    <main class="main-content"></main>

    <div class="other-content"></div>
  </div>

  <footer class="footer"></footer>
</body>

```


file:///Users/treta/Documents/Programmer/cra333/repos/webcom-talky/index.html

Responsive ▾ 376 x 652  DPR: 2 ▾ No throttling ▾ 



.header

.intro

.main-content

Desktop



Chrome

57



Opera

44



Firefox

52



IE

11*



Edge

16



Safari

10.1

Mobile / Tablet



iOS Safari

10.3



Opera Mobile

46



Opera Mini

No



Android

67



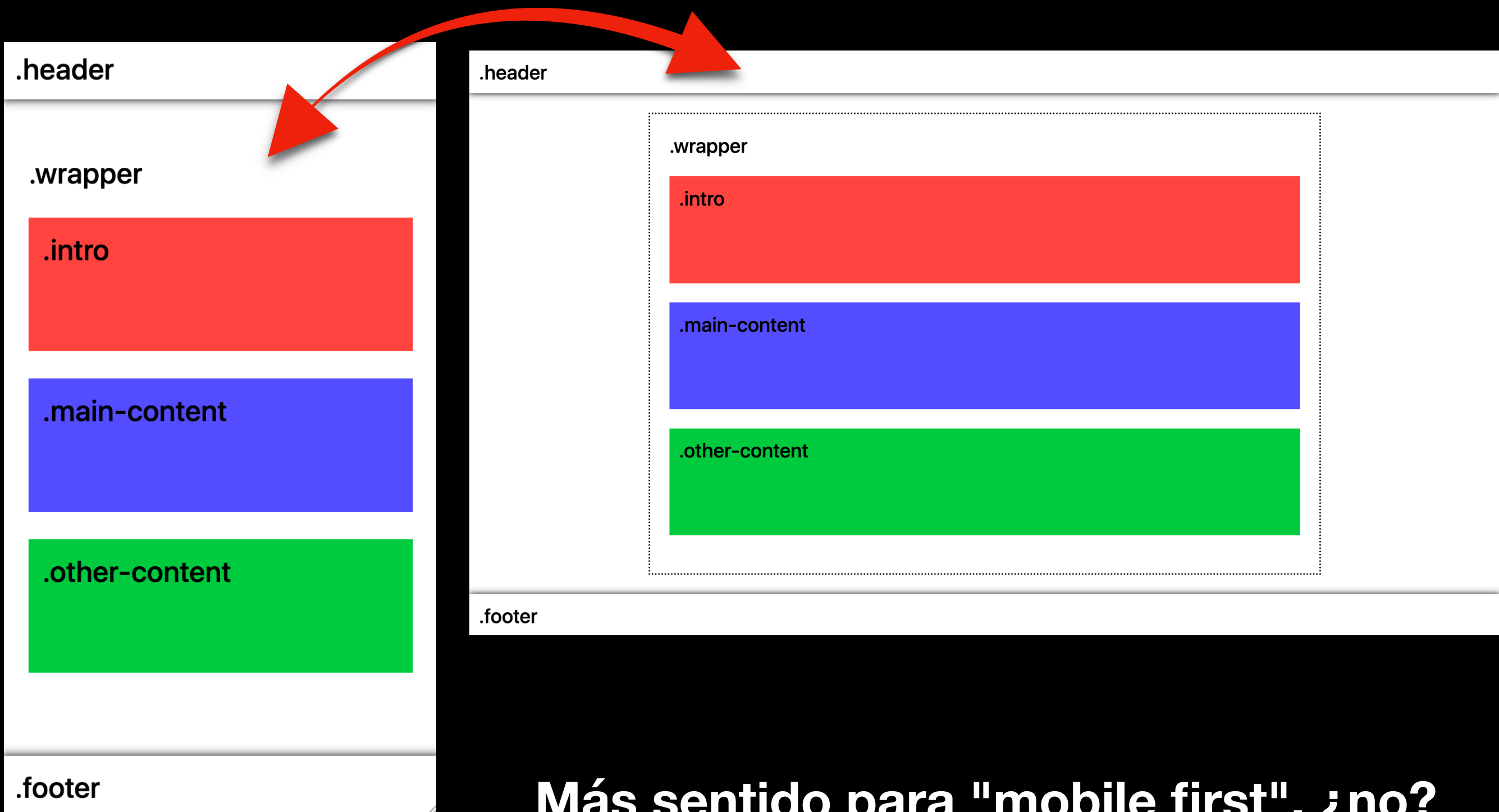
Android Chrome

73



Android Firefox

66



Más sentido para "mobile first", ¿no?

```
@supports(display: grid) {  
  @media(min-width:992px) {  
    .wrapper {  
      width: 992px;  
      display: grid;  
      grid-template-columns: 1fr 1fr;  
      grid-template-rows: auto 1fr 1fr;  
      grid-gap: 30px;  
    }  
  
    .main-content {  
      grid-column: 2/3;  
      grid-row: 2/4;  
    }  
  }  
}
```

```
/* IE Support */  
@supports(grid-area: auto) {  
  @media(min-width:992px) {  
    .wrapper {  
      width: 992px;  
      display: grid;  
      grid-template-columns: 1fr 1fr;  
      grid-template-rows: auto 1fr 1fr;  
      grid-gap: 30px;  
    }  
  
    .main-content {  
      grid-column: 2/3;  
      grid-row: 2/4;  
    }  
  }  
}
```

¿Cuándo es bueno pensar en CSS grid?

-> Siempre que estén por desarrollar un layout

¡Bienvenidos al futuro!



¡Muchas gracias!

Make CSS great again