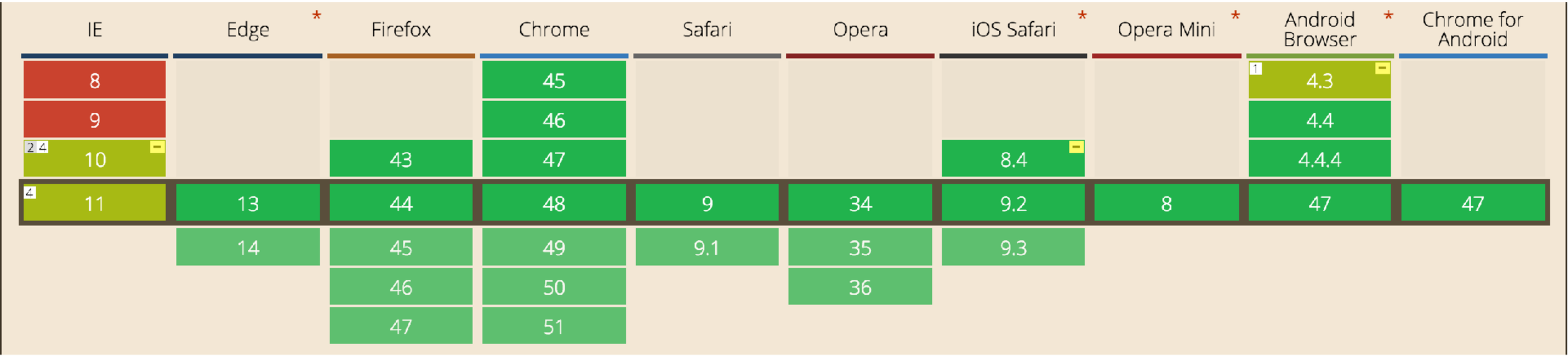
FLEXBOX

What is CSS Flexbox?

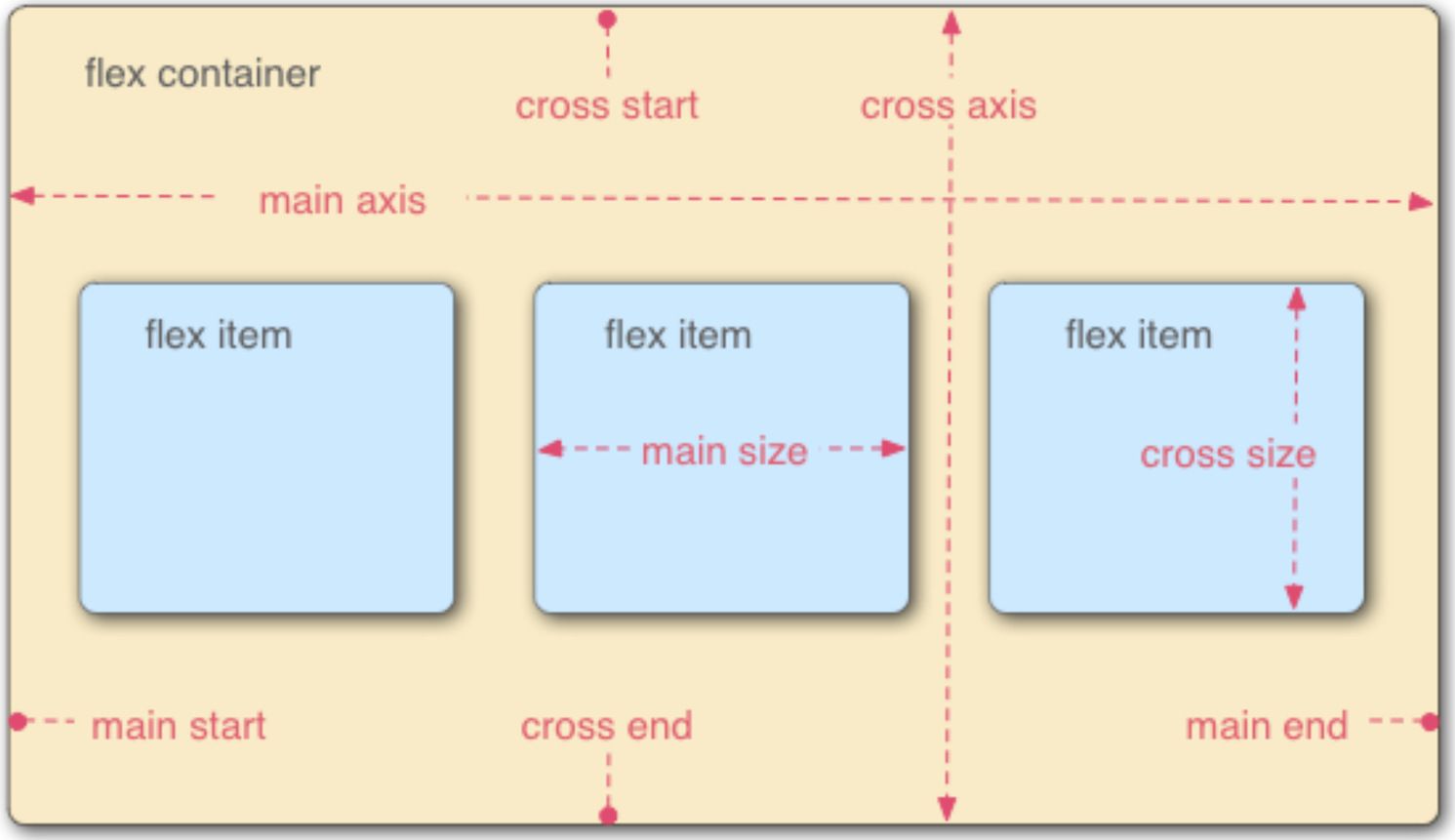
- The Flexbox Layout module aims at providing a more efficient way to lay out, align, and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").

Can I use it?

caniuse.com



The Container



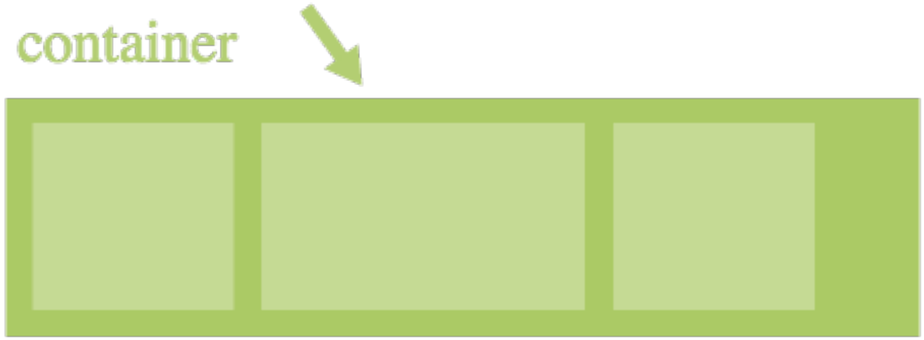
Flexbox Container: display

- display: flex defines an element as a flex container and enables a flex context for all its direct children.

```
.container {
  display: flex; /* or inline-flex */
}
```

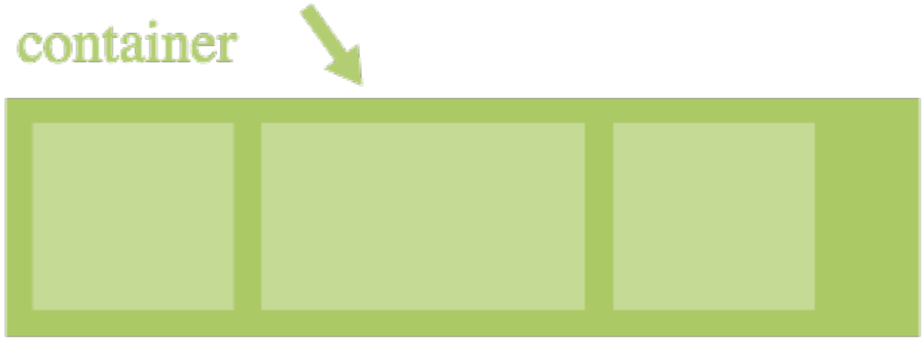
Flexbox Container

- Flexbox gives the container the ability to alter its items' dimensions (and order) to best fill the available space.



Flexbox Container

- A flex container expands flexible items to fill free space, or shrinks them to prevent overflow.



Flexbox Container: flex-direction

- Flexbox is (aside from optional wrapping) a single-direction layout concept. Think of flex items as primarily laying out either in horizontal rows or vertical columns.
- “main axis” and “cross axis”

Flexbox Container: Direction

- Think of flex items as primarily laying out either in horizontal rows or vertical columns.

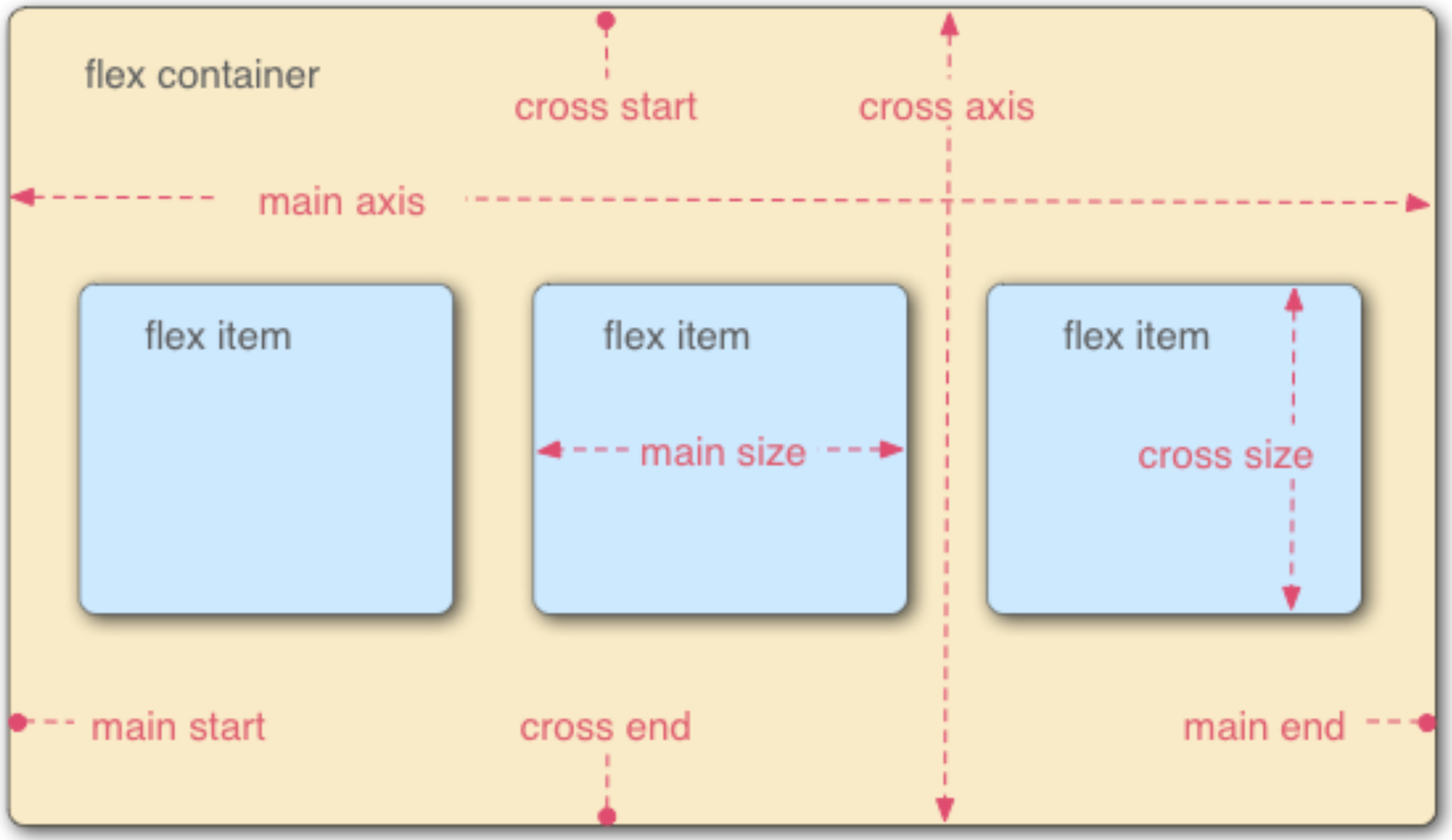

```
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

Flexbox Container: wrap

- Items will all try to fit onto one line. Items can wrap as needed with this property. Direction also plays a role here, determining the direction new lines are stacked in.

```
.container{
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

The Items


flex container
cross start
cross axis
main axis
flex item
flex item
main size
flex item
cross size
main start
cross end
main end

Flexbox Items: flex-grow

- Flex items can grow if necessary. This property accepts a unit-less value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up.


```
.item {
  flex-grow: <number>; /* default 0 */
}
```

Flexbox Items: flex-grow

- eg. if all items have flex-grow set to 1, every child will set to an equal size inside the container. If set to 2, that child would take up twice as much space as the others.


```
.item {
  flex-grow: <number>; /* default 0 */
}
```

Flexbox Items: flex-shrink

- This defines the ability for a flex item to shrink if necessary. Negative numbers are invalid.
- Not that necessary

```
.item {
  flex-shrink: <number>; /* default 1 */
}
```

Flexbox Items: flex-basis

- Like width property (or height, depending on flex-direction).
- If a relative value, indicates proportion of that item's width that should be applied.
- Default size of element before flex-grow or flex-shrink kick in

```
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

Flexbox Items: flex

- This is the shorthand for flex-grow, flex-shrink and flex-basis combined. The second and third parameters (flex-shrink and flex-basis) are optional. Default is 0 1 auto.

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

<liveCode />

RESPONSIVE

RESPONSIVE


I heard you want to be a web developer
Here are a few devices to test your site

RESPONSIVE DESIGN

RESPONSIVE DESIGN

- Website is fully functional for all screen sizes, resolutions and orientations

RESPONSIVE DESIGN

- Website is fully functional for all screen sizes, resolutions and orientations
- Born out of necessity (see previous slide)

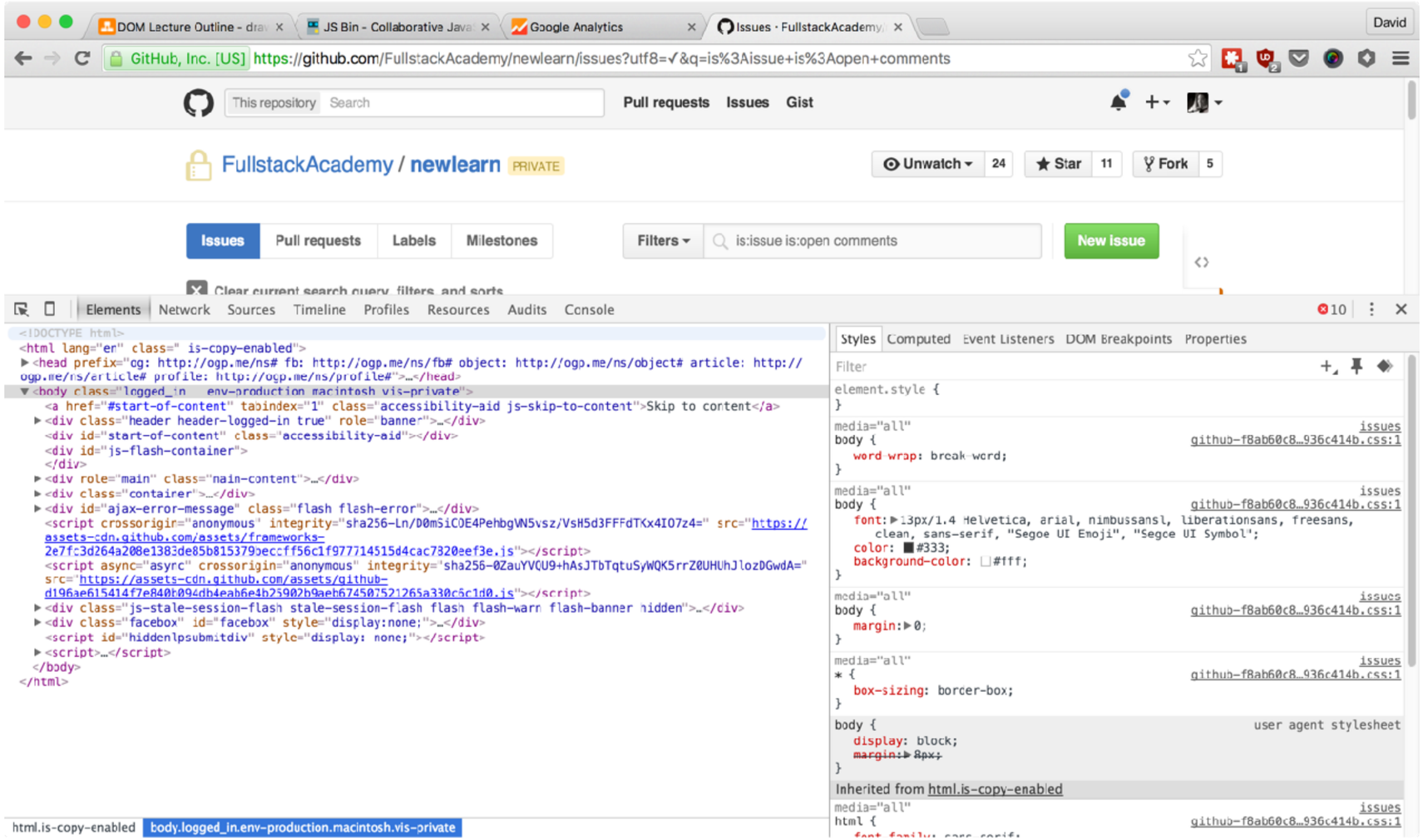
RESPONSIVE DESIGN

- Website is fully functional for all screen sizes, resolutions and orientations
- Born out of necessity (see previous slide)
- Developers and designers should cater to the user's environment, not the other way around

RESPONSIVE DESIGN

- Website is fully functional for all screen sizes, resolutions and orientations
- Born out of necessity (see previous slide)
- Developers and designers should cater to the user's environment, not the other way around

Developer Tools



<liveCode />