IT School®
FROM ZERO TO HERO!

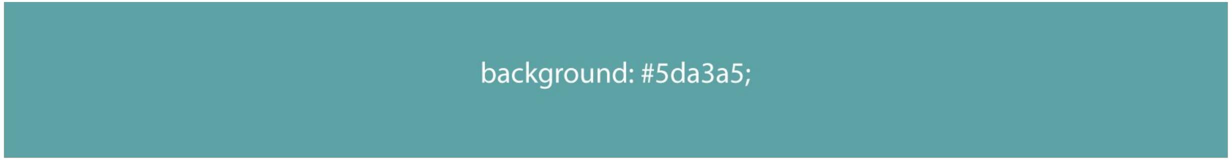
CSS Basic Rules - Part#3

Sesiunea 8 - 17/05/2021

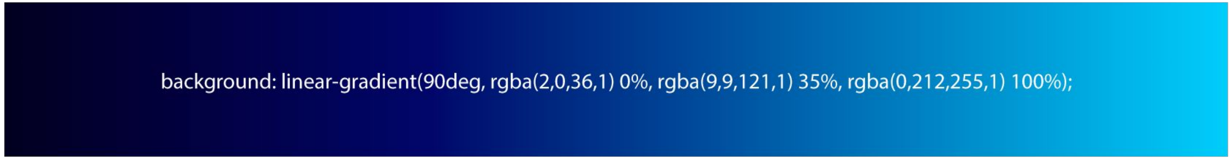
/menti.com\

/Recap\

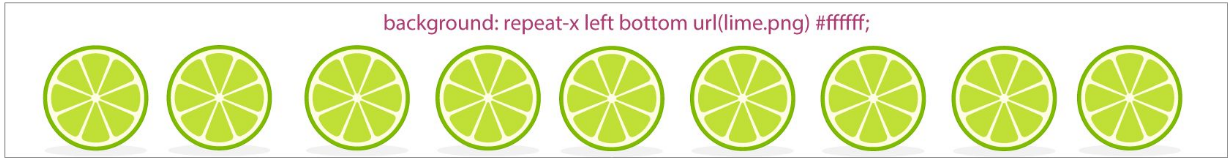
CSS: borders, backgrounds and shorthand props



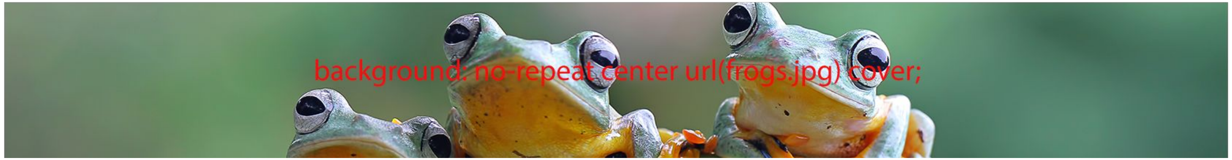
```
background: #5da3a5;
```



```
background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,9,121,1) 35%, rgba(0,212,255,1) 100%);
```

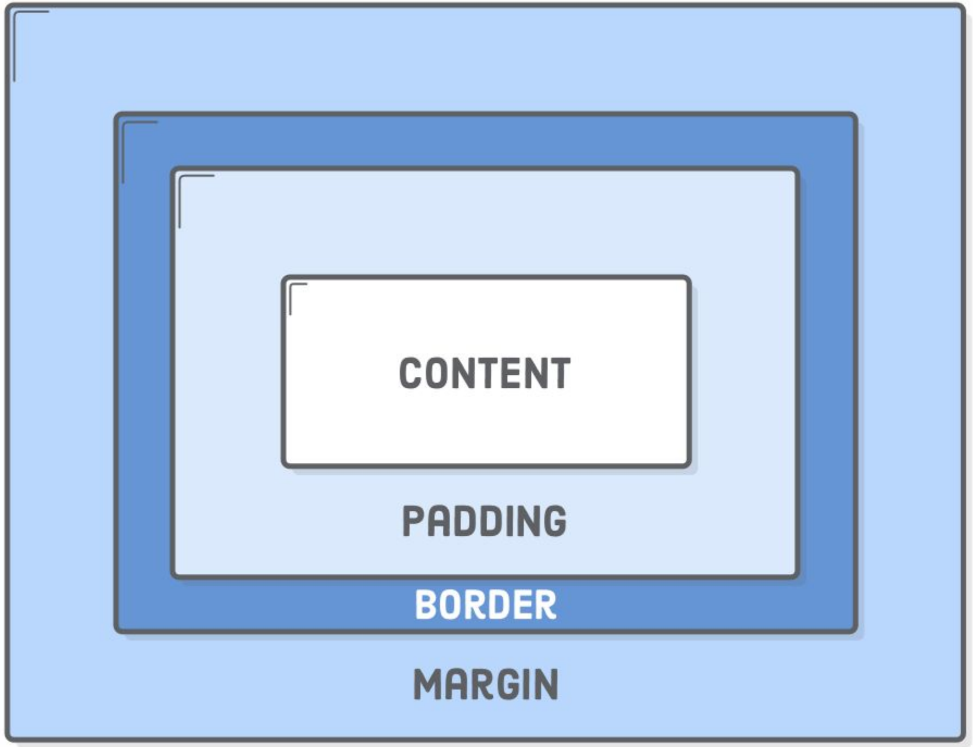


```
background: repeat-x left bottom url(lime.png) #ffffff;
```



```
background: no-repeat center url(frogs.jpg) cover;
```

CSS: box model, margins, paddings and box-sizing

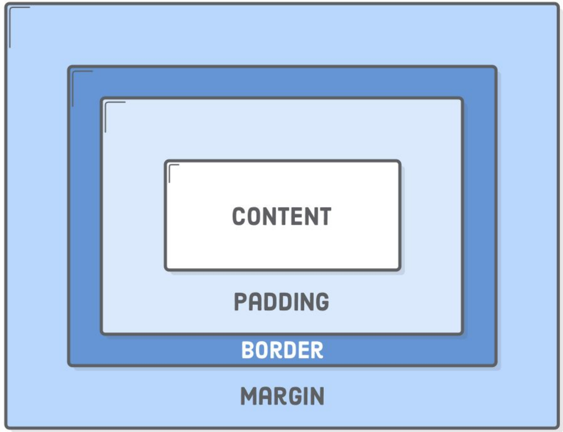


CSS: box-sizing

box-sizing

- content-box (default): the width and height props include only the content, excluding padding, border and margin
- border-box: the width and height props include the content, padding and border, but do not include the margin

Note: very useful on building your layout.



CSS: layout - position

position: (manipulate the location of an element)

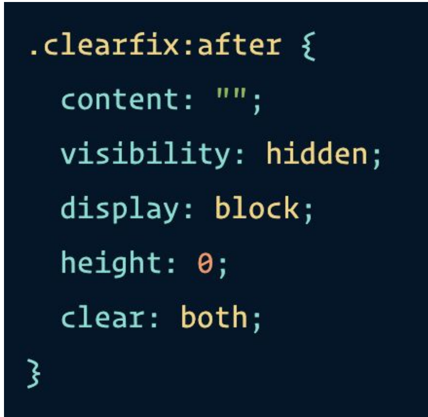
- static (static)
- relative
- absolute: relative to the first positioned parent
- fixed: relative to the document, not affected by scrolling
- **.relative-element .absolute-element {}**: positions element considering the relative parent
- all except static give access to **top, right, left, bottom, z-index** props

CSS: layout - float/clear

float: none|left|right|inherit (float to the specified direction of the container)

clear: none|left|right|both|initial|inherit (don’t allow floating to the specified direction of the specified element)

- floats can affect elements that contain them
- it can affect semantic web to be achieved (adding empty divs)
- .clearfix all time solution



```
.clearfix:after {
  content: "";
  visibility: hidden;
  display: block;
  height: 0;
  clear: both;
}
```

/Good practices\

Layouts: good practices

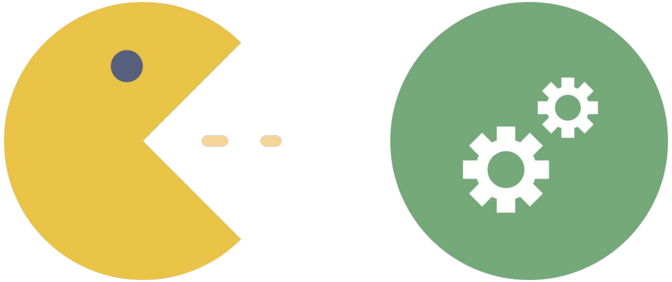
- Semantic Web: both HTML and CSS (classes, ids)
- Risky having styles directly on the <body>
- Readability is very important
- Positioning (relative, absolute), margins and paddings is not meant for layout building
- Avoid using <br> and <hr> elements for presentation purposes (alternative: decorative elements)
- We are aiming to write dynamic code (the less manual the better)
- Open to change, the way we know how to do it or they way it works might not be the best
- At all times we need to think of **adaptive layout**

Layouts: useful techniques

- Horizontal alignment

margin: 0 auto;

- Use an image wrappers



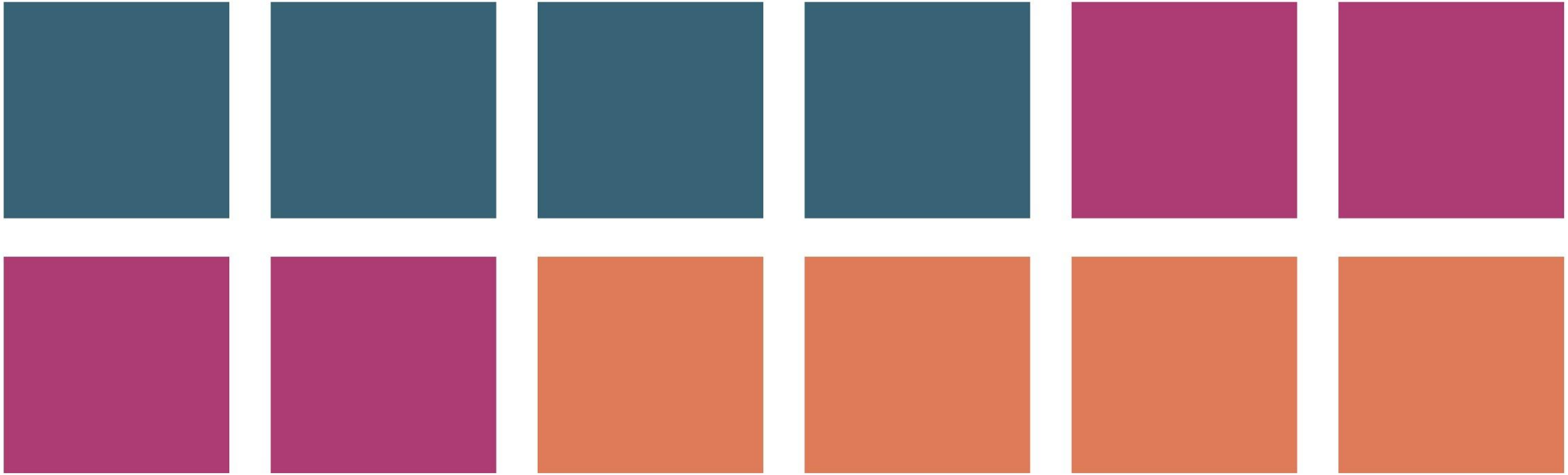
/code\

/Hands on: Floats\

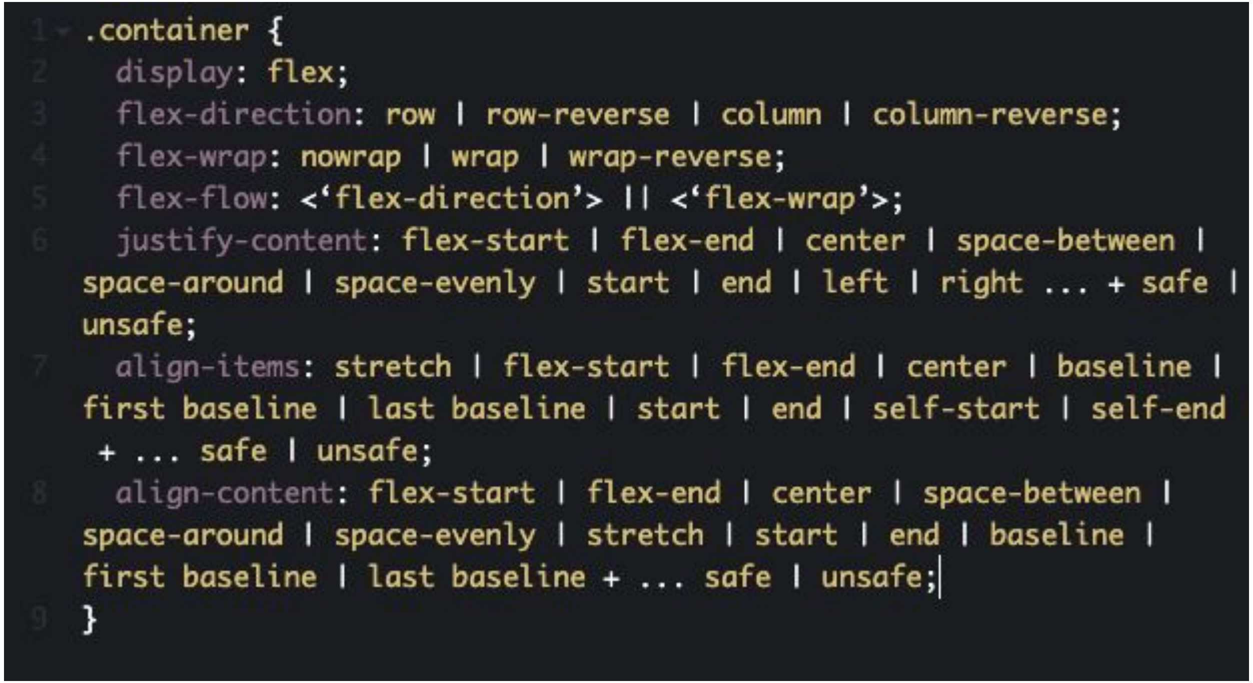
/shorthand props\

CSS: layout - float/clear

Exercise: using float/clear/clearfix rules place the squares of the same color on different rows

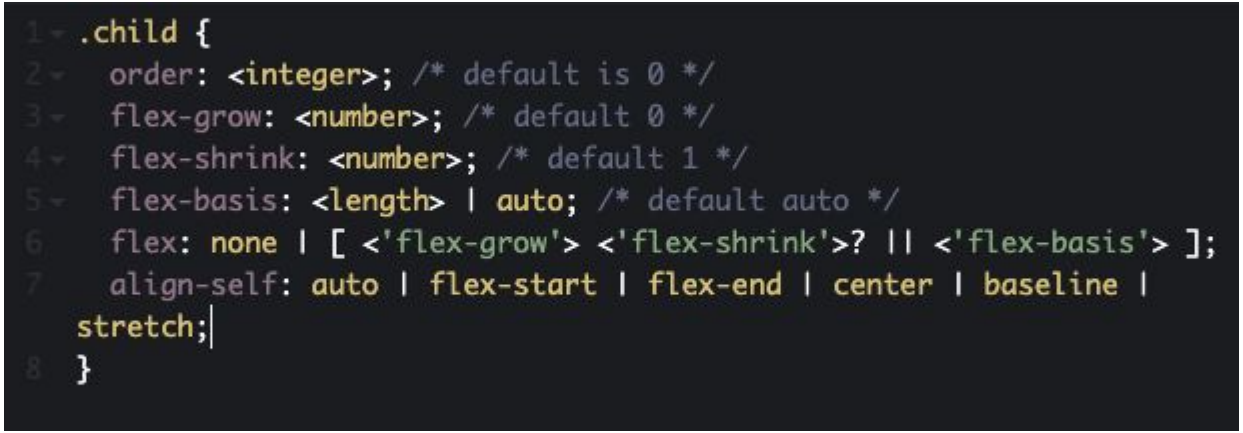


CSS: layout - flex



```
.container {
  display: flex;
  flex-direction: row | row-reverse | column | column-reverse;
  flex-wrap: nowrap | wrap | wrap-reverse;
  flex-flow: <‘flex-direction’> || <‘flex-wrap’>;
  justify-content: flex-start | flex-end | center | space-between |
space-around | space-evenly | start | end | left | right ... + safe |
unsafe;
  align-items: stretch | flex-start | flex-end | center | baseline |
first baseline | last baseline | start | end | self-start | self-end
 + ... safe | unsafe;
  align-content: flex-start | flex-end | center | space-between |
space-around | space-evenly | stretch | start | end | baseline |
first baseline | last baseline + ... safe | unsafe;
}
```

CSS: layout - flex



```
.child {
  order: <integer>; /* default is 0 */
  flex-grow: <number>; /* default 0 */
  flex-shrink: <number>; /* default 1 */
  flex-basis: <length> | auto; /* default auto */
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ];
  align-self: auto | flex-start | flex-end | center | baseline |
stretch;
}
```

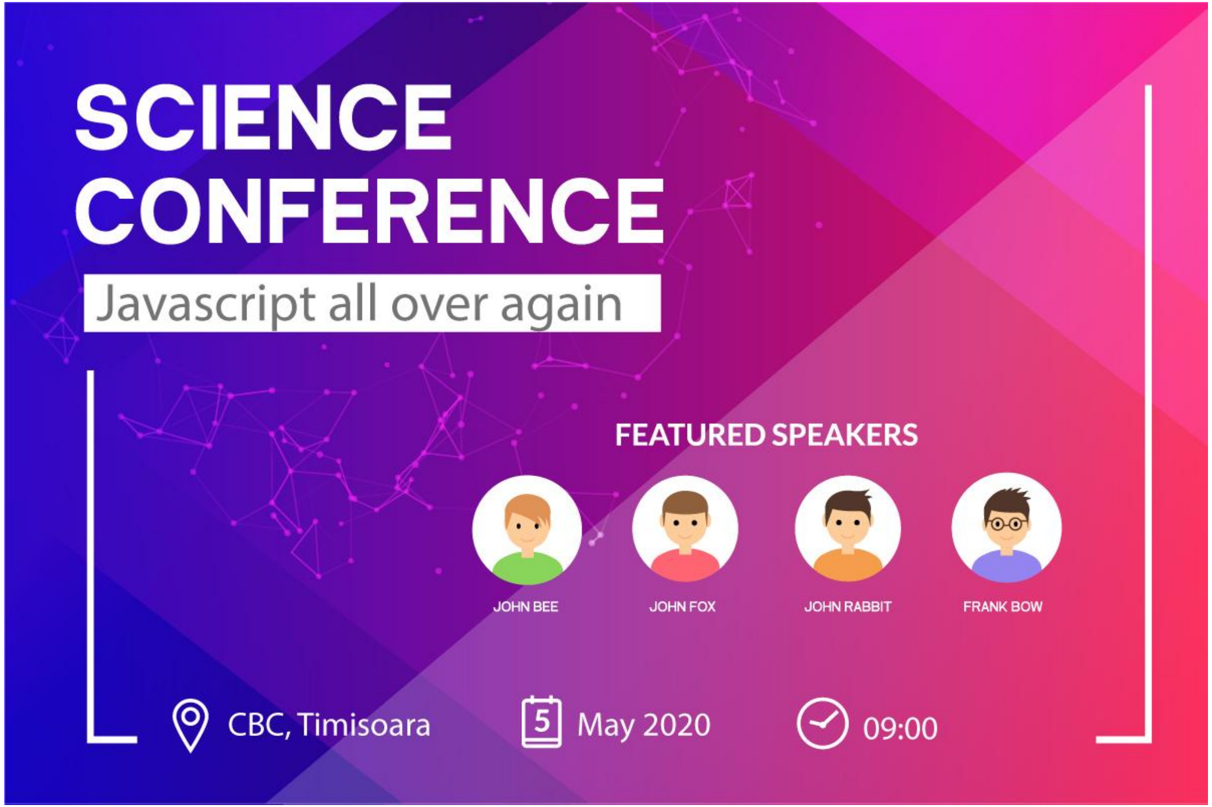
IT School®
FROM ZERO TO HERO!

CSS: layout - flex

Exercise: Let’s play a game

/Practice = improvement\

Assignment #2



- Well-formed doc
- Semantic tags
- External stylesheet
- Use flex for positioning

IT School®
FROM ZERO TO HERO!

/Q&A\

Resources

- MDN Web docs: https://developer.mozilla.org/en-US/
- W3Schools: https://www.w3schools.com/
- A game for learning CSS flex: https://flexboxfroggy.com/#it
- Flex cheat sheet: https://www.freecodecamp.org/news/flexbox-the-ultimate-css-flex-cheatsheet/
- A complete guide to flexbox: https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- Can I Use: https://caniuse.com/
- CSS reset: https://gist.github.com/DavidWells/18e73022e723037a50d6

IT School®
FROM ZERO TO HERO!

Thank you

Next: Intro to Git