# CSS

## Inheritance

Some properties are inherited by default no matter how deep nested elements

<div id="box">

    <section>

        <section>This is some text inside the box.<section>

    </section>

</div>

#box {

      background-color: brown;

      padding: 100px;

}

“This is some text…” will also have background color brown. The inheritance works on those properties for all selectors: tags, classes, id.

Some properties like `padding` are not inherited, to inherit them it should be set to `inherit`

#sec {

  padding: inherit;

}

But it inherits from close parent, not from grandparents:

        <div id="box">

            <section>

                <section id="sec" >This is some text inside the box.<section>

            </section>

        </div>

#box {

      background-color: brown;

      padding: 100px;

}

#sec {

  padding: inherit;

}

The padding of #sec will be 0px because the padding of parent <section> is default 0px.

## Select needed element

To select all descendants (children, grandchildren etc.) write

.class \* {

  padding: 10px;

}

To select only first-level children:

.class > \* {

  padding: 10px;

}

Select first elements B directly followed by A elements:

A + B {

  padding: 10px

}

Select all elements B followed by A element (on the same level of nesting):

A ~ B {

  padding: 10px

}

Select first child(or grandchild, etc.) of element:

div p:first-child{

  padding: 10px;

}

Select every element that is the 8th child of another element:

:nth-child(8){

  padding: 10px;

}

## FlexBox

In flex-box the `flex` property works bad:



It’s ignore 200px size.

So I prefer to use related parameters:

#div1{

  background-color: coral;

  width: 100%;

  flex-shrink: 1;

}

#div2{

  background-color: lightblue;

  width: 200px;

  flex-shrink: 0;

}

#div3{

  background-color: lightgreen;

  width: 200px;

  flex-shrink: 0;

}

The div1 take all the rest place and only div1 is shrink.

Don’t use flex-grow and width.

#div1{

  background-color: coral;

  min-width: 100px;

  width: 50%;

}

#div2{

  background-color: lightblue;

  min-width: 100px;

  width: 100%;

}

#div3{

  background-color: lightgreen;

  width: 200px;

  flex-shrink: 0;

}

div2 is twice wider than div1, but if shrink – 100px both:

## Width Height 100%

If ancestors heights are set to 100% - the elements will take the whole screen space.

But, if one of nested element are bigger than screen size, and the `overflow` of it parent is not set – it will EXPAND the screen size, and all ancestors will take the 100% of new screen size. This is because the default overflow is `visible`.

Also setting h/w in % works like shit, so try to use these parameters together:

**flex-grow: 1** – sets that the element can grow and how much relatively to other growing elements,

**flex-basis: 100px** – sets the minimal size (height if the parent is flex-direction column)

**overflow: hidden** (or another) ­– limits the size of children.

Then, when you have a growing box, if you put another element inside – you can operate on **max-width max-height** parameters.

# HTML

## Form input validation

There are a lot of input type like “text”, “password”, “number”, “tel”, “time”

`accept` set file filter when the file explore window is opened, but the user can select any file still.

<label> is binded to `id`.

<input> is binded to controller method parameters through `name`.

Additional fast validation in js:  
  
html and js validation is only on client side and can be avoided. To enhance security add server-side validation:

