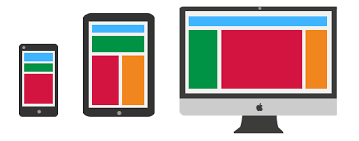
{ CSS Training }

*#4*

*Responsive design*

Device Independent Pixels



Hardware pixels = 2560px

Device Independent Pixels (DIP) = 1280px

Device pixel ratio = 2 (hardware / DIP)

Calculating CSS pixels

1920x1080 screen , pixel ratio=2 -> 960 pixel is wide the viewport

(can be rounded)

Viewport meta tag = we instruct the page to match the screen’s width in DIP

<meta name="viewport" content="width=device-width, initial-scale=1">

Use relative positions rather than absolute positions such as width 100%

max-width: 100%;

touching target

our finger is 10mm wide = 40 css pixel

buttons, links – 48x48 px at least (min-width, min-height)

Prioritize content

mobile first, bigger next

What’s the most important thing I want to see on the screen? When starting big and then designing down, it can be too easy just to hide content that may be (un)important.

Instead, priorotize content and work up – the key content is always on the page and users get the full experience no matter what device they are using. Not to mention performance: I’m forced to think about performance from the beginning (how much data I send... etc).

Media query

A responsive website, changes based on the characteristics of the device (and needs to apply different styles).

<link rel="stylesheet" href="xy-medium.css" media="screen and (min-width:500px)">

or

@media screen and (min-width: 500px) and (...) { ... }

or

@import url(„xy.css”) only screen and (min-width: 500px);

(avoid this because of performance reasons)

min-device-width = screen size

min-width = browser width

The right point when the page changes layout is called a break point.

Find the right breakpoint by looking the content and not the device.

Patterns

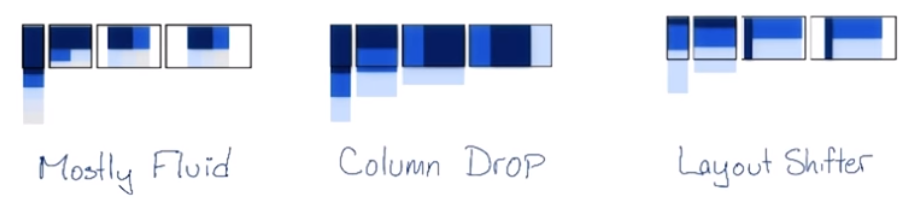
grid fluid system, flex pattern...

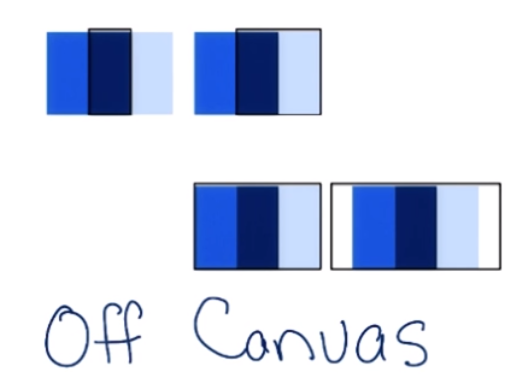
#flexcontainer {

display: flex;

flex-wrap: wrap;

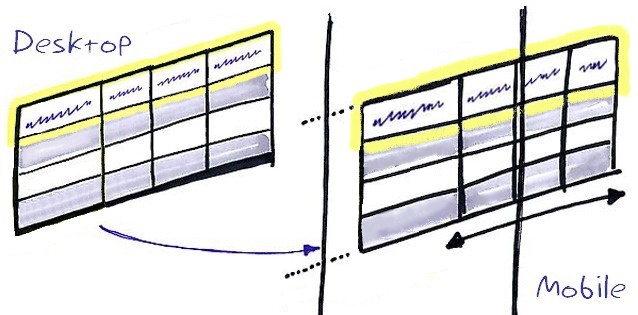
}





Off canvas navigation = hamburger menu

Responsive tables



If there are more than a few columns, there is a chance to overflow the viewport on smaller screens, forcing horizontal scrolling.

Solutions:

* Hidden columns
* Technique called no more tables
* Contained tables

Different content needs different solutions.

Hidden columns hides columns based on their importance as the viewports size gets smaller. Biggest problem is hiding information from the user.

@media screen and (max-width: 499px) { .hidden { display: none; } }

No more tables technique: below a certain viewport width the table is collapsed and resembles a long list, as opposed to a table data. All data is visible, scrolling is vertical. Every column has essentially become its own row with the table header beside it.