**WEB DESIGN**

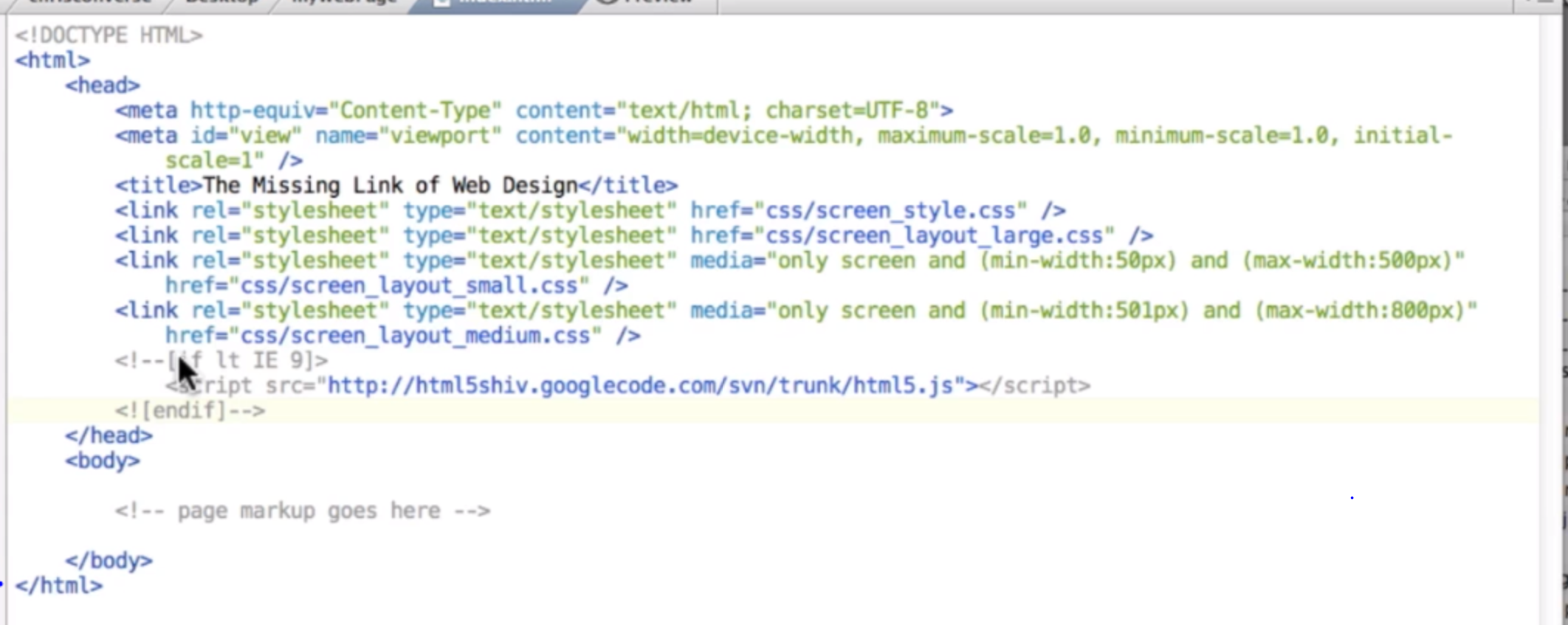
Small screen-> approx. 50 – 500

Medium -> approx.501 – 800

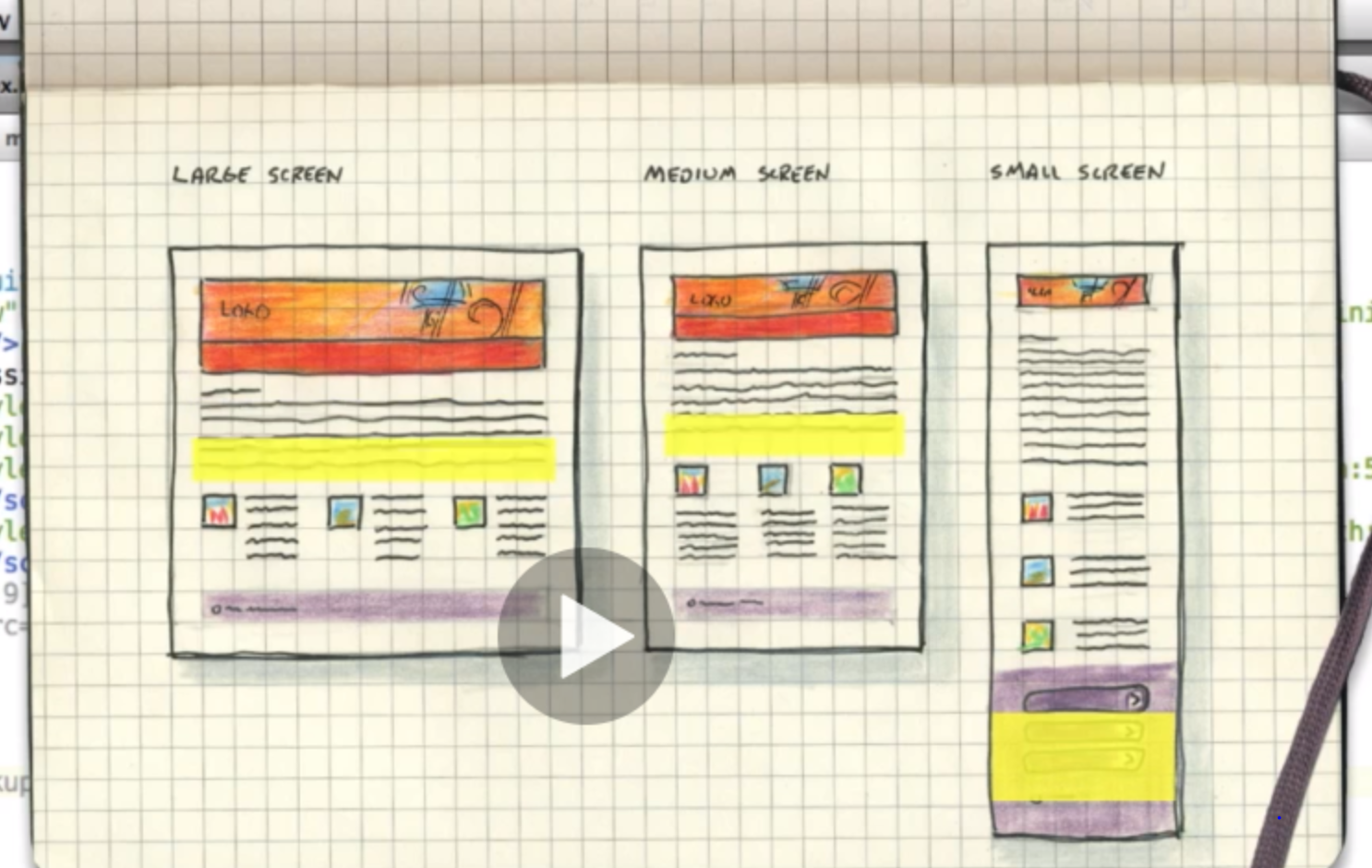
* Meta view port for non scalable
* [If lt IE 9]<script>…[EndIf]
* Lock after 980px so it wont stretch further

<header><article><div><nav><footer>

* Skeletal



* Visualizing



* Repeat-x
* Margin: auto
* .page -> relative -> anything absolute inside page will be relative to .page
* Max-width -> won’t stretch after that
* Compound css -> class = “promo one” .promo.one ->.promo p (here p is child)
* Clear -> clear float on both left right

Conquering Res. Layout

* padding [<-> Item <->]
* div p -> selects p inside div 🡪 different ways of css selections, lookup online
* width-> 80% of parent
* height-> X padding🡪 use 8em 🡪 16\*8 px instead

EM/REM

EM

* em -> font size/padding/margin/width/heights
* In relation to parents
* Compounds of parents
* Fontsize looks at parent , rest looks at the fontsize of that element

REM

* To fix compounding
* Looks at html/root
* Consistent

# Stick to rem for font size

* Transition, gradient…see

# Padding/Margin use EM so that it is consistent with font size

* Don’t let text flow all the way, use max-width: 750px

CSS,VH,VW,Vmax,Vmin

\*{box-sizing:border-box} so that width/height is exact without padding/border addition

\* vh-> view port height , of the whole screen. 80vh -> 80% of vh sometimes content is longer so it overflows

\* width:80vw

\* height:20vmax, whichever is maximum…height/width 🡪 vmin

\* can use for padding🡪 padding:10vh 0

\* also font-size:8vw -> not recommended as shrinking and extending makes too small or too big vmin works a little better



* Use more on height

##FLEX

Container> row> col col col

.row{display:”flex”} !! creates a flex container, flex-direction:row(default)

.flex shrinks down to smallest possible

.adding space between columns

. gap in .row 100px gap: 100px (only in firefox right now)

.other

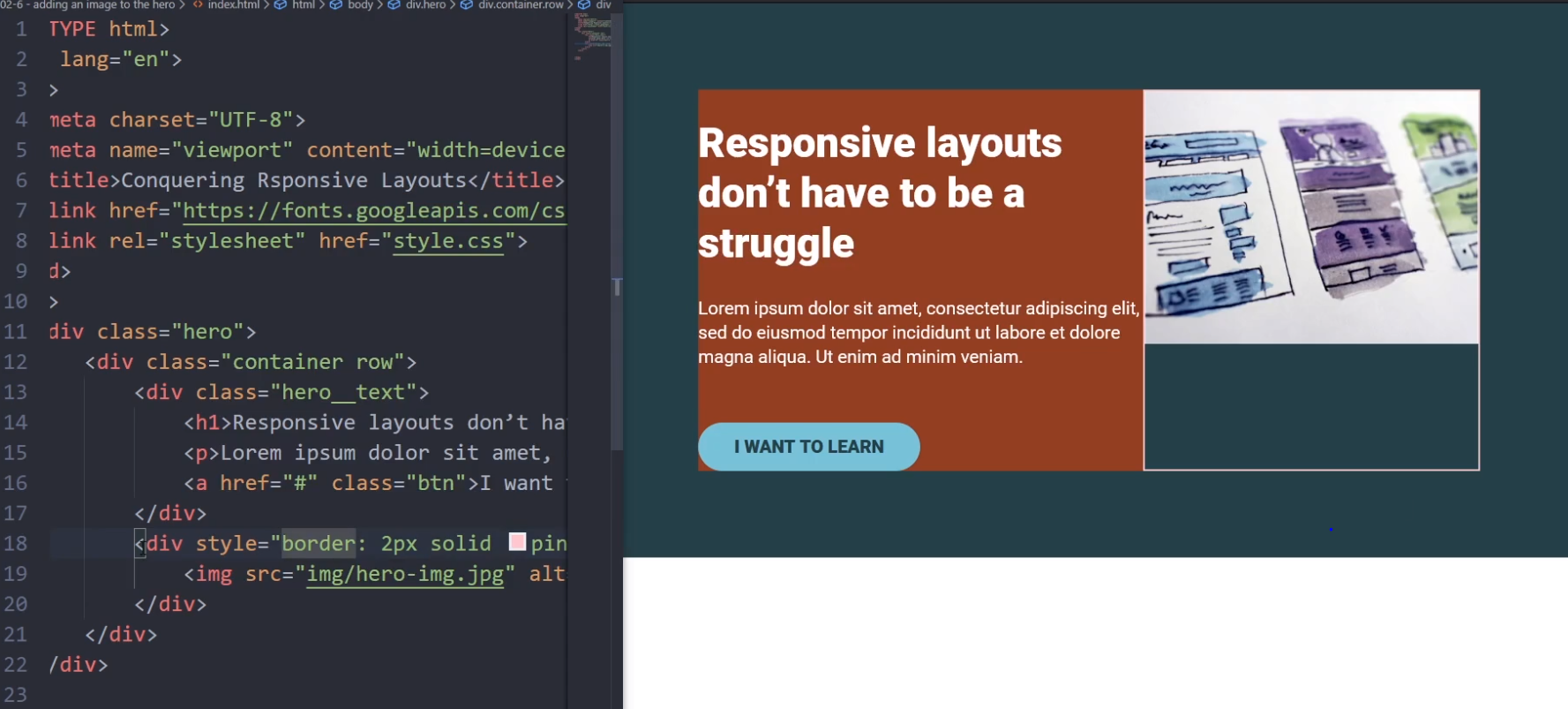
.col+.col{

background-color:limegreen;

margin-left:100px;

}

.reducing html-> Classname compounding



To avoid stretching image than real size:::

* In Classname ‘container row’ 🡪 row{display:’flex’} If img kept without div, img stretches
* In row, flex—align items: flex-start to display in similar way (at top)
* Or img without div can have a class with prop in css {align-self:’flex-start’}

Same scenario :: Above hero\_text has width of 60% and image has nothing, image shrinks to smallest possible and text becomes longer :: to avoid that

* Width:100% (not relevant~)in both divs but what if we want text to be 60%, set width for all flex children
* Max-width of image 100% so it doesn’t stretch too much that it looks ugly

But shrinks though

* Justify-content states how flex children are to be distributed in flex container

space-between makes space between children

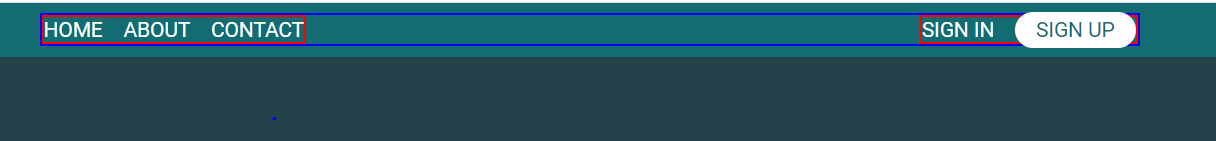
* To avoid images side scrolling :: max-width: 100 % within div class
* Specify width in flex children
* To push certain contents to right separate with class and, flex justify-content:space-between. Also as below:

.nav\_\_item--push-right {

  margin-left: auto;

}

* TO make this



Code:

.nav\_\_item + .nav\_\_item {

  margin-left: 1em;

}

* Space in nav

width:100% or flex-grow:1(enabled) flex-shrink:1(mostly)

Media Queries

Desktop first here then media queries for mobile design

* Issue things shrinks down, stacking is better
* Media Queries below to override things
* Divide things to row and in mobile design .row:block, width:100% on other

Stuffs that doesn’t occupy all of the screen

* Add Margin where necessary



* Breakpoints
* Use responsive and set breakpoints when things start to look weird
* Use less number of breakpoints as far as possible
* Meta tag viewport responsible for making site responsive else it will shrink the whole page down

details, summary, line spacing, letter spacing, line height…

* Min-height in px works or vh works so that it can adjust according to content
* Width in % or max-width min-width in px or ems
* Grid