SUMMARY OF COURSE

CONQUERING RESPONSIVE LAYOUTS BY KEVIN POWELL

RATINGS: 4/5

SECTION 1 – INTRODUCTION

UNIT1: Intro & Why the course is formatted in this way

- 21 days’ challenge
- Time to absolve the lessons and practice it
- Make Responsive layout as easy as possible

UNIT2: My editor and a useful extension

- He uses VSCode
- He uses “Live Server” extension

UNIT3: When the next content will be available

- An email to let us know when the next content will be available

UNIT4: Asking Questions

- Join a Discord community
- Using” ``` **** ``` ” 3 backticks to type code on discord

SECTION 2 – DAY 1: USING PERCENTAGES AND AVOIDING HEIGHTS

UNIT1: Percentage vs Fixed Widths

- By default, most things are responsive in HTML and CSS, we affect it when we start assign come values
- “margin: 0 auto”: to center it
- Using “***%” instead of “***px” brings back the responsiveness

UNIT2: Percentages on the child

- “overflow: hidden”
- “50%”: it means 50% of the parent class

UNIT3: Why it’s a good idea to avoid heights

- Avoid using “height”, we can use paddings instead
- “em” & “rem”???: https://youtu.be/_-aDOAMmDHI

UNIT4: Challenge #1-Starting Files

- Challenge-01

UNIT5: Challenge #1

- Fix the text placement
- Solutions will be up in 2days

SECTION 3 – DAY 3: ENTER MAX-WIDTH

UNIT1: Challenge #1 – Solution

- I got it right "..smiles"

UNIT2: Adding in a max-width

- "max-width": the limit it gets big to
- It is quite essential for big screens so that things don't over extend

UNIT3: Challenge #2 – Starting Files

- Challenge-02

UNIT4: Challenge #2

- Fix the responsiveness applying "max-width"

UNIT5: Some notes on the challenge

- Allowed to modify the HTML

SECTION 4 – DAY 4: EXTRACURRICULAR ACTIVITIES

- https://www.youtube.com/watch?v=IWFqGsXxJ1E
- "VH/VW": Viewport Height and Viewport Width

SECTION 5 – DAY 5:PRACTICE TIME

UNIT1: Challenge #2 – Solution

- "Shift + Ctrl + P": nice shortcut
- "width": should usually be a % value
- "max-width": should usually be a "px" value
- Better solution in screenshot

UNIT2: Challenge #3

- Figma Sketch download: https://www.figma.com/file/dGPcL6Ym2avOt4736aHb2B/conquer-responsive-layout-week-1-final-challenge?node-id=0%3A1
- Roboto fonts

UNIT3: The Challenge

- Replace the figma link with a pdf that outlines the font info and colors
- No copy and pasting, write everything from scratch

UNIT4: Challenge-design-specs pdf

- challenge-design-specs.pdf

SECTION 6 – DAY 6: IT'S THE WEEKEND

UNIT1: Review of the first week

- Links to some extra reading, they will supplement what we are covering here

UNIT2: Why you shouldn't use "em's" for font-size

- "em": ems build on each other, getting there font by multiplying the parent font to get their own fonts
- Its better to use "rem" than "em" because "em" are compounded
- "rem" make use of the root only, so they are more manageable, default font size is usually 16px
- You can use "em" for paddings an margins, they use the font size of the text

UNIT3: A Tale of width and max-width

- https://css-tricks.com/tale-width-max-width/

SECTION 7 – DAY 7: SOLUTION TO CHALLENGE #3

UNIT1: My solution to challenge #3

- Emmet documentation: https://docs.emmet.io/
- Box-sizing: border box> https://youtu.be/WlGQdgy-M6w
- The more organized we are the easier to make to make it responsive
- "em" for margin and padding are relative to the font size
- "hover" and "focus"
- Good to use "Box-sizing: border box" it removes unnecessary complexity

UNIT2: Challenge-finished.zip

- Downloaded “challenge-finished.zip” > His solution for Challenge 3

SECTION 8 – DAY 8: FLEXBOX BASICS

UNIT1: Flexbox Basics

- Flexbox was the first actual layout property given to us in CSS

UNIT2: An Introduction to flexbox

- “display: flex”, flex container are the parents while flex items are the children
- “div” by default have a width of 100%, flex items try to be as small as possible

UNIT3: Adding space in between columns

- “gap: 100px” > it is only supported for firefox
- “+”: combinators, “.col + .col”

UNIT4: Flexbox Challenge #1

UNIT5: Flexbox Challenge #1 – Design Specs

- Download “02-03-challenge-layout.pdf”

UNIT6: Flexbox Challenge #1 – starting files | HTML & CSS

- Download “challenge start.zip”

UNIT7: Flexbox Challenge #1 – starting files | Codepen

SECTION 9 – DAY 9: A DEEPER DIVE INTO FLEXBOX

UNIT1: My Solution to Flexbox challenge #1

- He is a big fan of everything having a class name
- My solution was very identical to his own…hurray

UNIT2: Reducing the amount of HTML needed

- You can give a “div” two class names to reduce number of divs

UNIT3: Adding a hero image

- Putting the image in a “div” stretches the “div” not the image
- “align-self: flex-start”

UNIT4: Column widths and flexbox

- “Justify-content: space-between”, most commonly used, works if empty space is available

UNIT5: Flexbox challenge #2

UNIT6: Ensuring the image is responsive

- Img {max-width: 100%}: to make images responsive

UNIT7: Flexbox challenge #2-Design Specs

- Download “02-05-challenge.pdf”

UNIT8: Flexbox challenge #2 – starting files | HTML & CSS Files

- Download “02-6-challenge 2.zip”

UNIT9: Flexbox challenge #2 – starting CodePen

SECTION 10 – DAY 10: A DEEPER DIVE INTO FLEXBOX

- https://www.youtube.com/playlist?list=PL4-IK0AVhVjMSb9c06AjRlTpvxL3otpUd

SECTION 11 – DAY 11: USING FLEXBOX FOR A NAVIGATION

UNIT1: Flexbox Challenge #2 – Solution

- It is good to use “semantic HTML”
- His solution quite similar to mine

UNIT2: Flexbox Challenge #3

- Do a navigation bar

UNIT3: Flexbox challenge #3 – starting files | HTML & CSS Files

- Download “nav_challenge.zip”

UNIT4: Flexbox challenge #3 – starting files | Codepen

UNIT5: Flexbox challenge #3 – solution

- “margin-right” on nav items
- “put a container on the whole nav”

SECTION 12 – DAY 12: GETTING FANCY WITH OUR NAVIGATIONS

UNIT1: Fancier up our navigation with flexbox

- “margin-left: auto” > pushes to the right
- Giving things borders/background colors is a good way to learn (easy to visualize what is going on)

UNIT2: More improvement to the navigation

- Main axis and Cross axis
- “justify-content” works on the main axis while “align-items” works on the cross axis
- “flex-grow” and “flex-shrink” (will most always be 1)

UNIT3: Centering things the easy way

- “auto” margins are very awesome

UNIT4: Flexbox challenge #4

- Starting from a completely blank file
- It not enough to just watch tutorials you also have to practice
- If you get frustrated leave it and come back to it

UNIT5: Flexbox Challenge #4 – Design Specs and Images

- Download “flexbox-challenge-4.zip”

SECTION 13/14 – DAY 13/14

- Enjoy your weekend

SECTION 15 – DAY 15: INTRO TO MEDIA QUERIES

UNIT1: Media query basics

- We will try to simplify media queries
- They help us rewrite or overwrite existing styles upon certain conditions
- “@media (**media feature**) {}”
- Uses mostly “min-width/ max-width” for media features
- “Desktop/ Mobile First”
- They have to arranged sequentially, the order is important for “min-width” arrange from smallest to biggest

UNIT2: Adding one to our layout

- Suggest placing all the media queries at the bottom

- Decide if we are writing “max-width” or “min-width” first
- “display: block”

SECTION 16 – DAY 16: WHAT BREAKPOINTS TO USE?

UNIT1: How to decide what breakpoints to use?

- Breakpoint: the width are which changes occurs
- Breakpoint the point at which the layouts starts to fail
- Try to keep your breakpoint as minimum as possible
- https://www.freecodecamp.org/news/the-100-correct-way-to-do-css-breakpoints-88d6a5ba1862/

UNIT2: The 100% correct way to do CSS breakpoints

- Use 600px, 900px, 1200px, and 1800px if you plan on giving the giant monitor people something special
- Try to learn SASS

SECTION 17 – DAY 17: THE VIEWPORT META TAG

UNIT1: The meta viewport tag

- Helps our site to work well
- Never delete your meta viewport tag

SECTION 18 – CHALLENGE SOLUTION & MOBILE-FIRST

UNIT1: Flexbox Challenge #4 solution- Writing the HTML

- Writing the HTML using the Big Screen design, he prefers to do mobile-first for writing the CSS
- Best to have the design and code editor on split screen while writing the HTML (need a bigger screen for this)

UNIT2: HMTL-finished.zip

- Download “html-finished.zip”

UNIT3: Flexbox Challenge #4 solution- Writing mobile-first CSS

- Desktop-First is the most common, but he thinks Mobile-first is the better option
- “shift+ctrl_m” > on Mozilla to enter a responsive mode
- Starts with “box-sizing” , then “body: margin 0”

- “img: max_width = 100%” to make them responsive
- “h1 h2”: remove the “margin-top”=0, to prevent alignment issues
- The text should not be touching the sides
- Smallest devices are about “300+px”
- Starting with mobile first, you don’t really have to worry about layouts

UNIT4: Flexbox Challenge #4 solution- Adding media queries for larger screens

- Don’t allow your “text line” of task to run/stretch across on big screens, limit the max-width
- Try to limit the amount of breakpoints
- You might have to change the font-size for larger screens
- I think I will start with mobile-first from now on

UNIT5: Flexbox Challenge #4 solution – A look at min-height

- You can use “min-height” but not fixed height
- “lorem+number of words”
- Learn to make images responsive < set the image tag to “max-width: 100%”

UNIT6: Flexbox Challenge #4 finished.zip

- Download “finished.zip”

SECTION 19 – DAY 19: MOBILE NAVIGATION CHALLENGE

UNIT1: Navigation Challenge

- We will be working with his own files
- Styling the Nav bar

UNIT2: Navigation Challenge – layouts

- Download “navigation-challenges.pdf”

UNIT3: Nav-challenge.zip

- Download “nav-challenge.zip”

UNIT4: Navigation Challenge | Starting Codepen

SECTION 20 – DAY 20: MOBILE NAVIGATION CHALLENGE SOLUTION

UNIT1: Navigation Challenge | My Solution

- Remember always mobile first

SECTION 21 – DAY 21: YOU'VE DONE IT

UNIT1: You've done it

- Website are responsive before writing CSS
- Usually a "desktop first" is the culprit, go with "mobile first"
- Mastering anything isn't about learning as much as you possibly can, practice it to mastery
- We would have a last challenge
- Try "Free Code Camp", "frontendmentor.io"
- Keep pushing yourself, you will create amazing things

UNIT2: Links to other resources

- https://javascript30.com/
- https://www.freecodecamp.org/

UNIT3: Your Final Challenge

- Take a "mobile-first approach"
- Don't copy and paste any code, write it out
- You can use "Oswald" instead of "Antartician Headline"
- Use the breakpoints in the pdf as targets

UNIT4: Final Challenge | Design Specs

- Download "final-challenge.pdf"

UNIT5: The image used

- Download "the-image.jpg"