Comp 324/424 - Client-side Web Design

Fall Semester 2024 - Week 10

Dr Nick Hayward

CSS3 Grid - sample layouts

intro

- grid layout enables more complex and interesting layout options

 overlap, layers...
- sample layouts using CSS grid structure
 - common layout options and designs
 - useful repetition of design
 - modify base layouts for various site requirements
- sample layouts
 - responsive layouts
 - auto placement for dynamic content and media
 - platform agnostic designs
 - useful with SPA, SVG, async patterns &c.

HTML5, CSS, & JS - example - part 4

add flex to grid layout

- an additional option to consider flex layouts
 - aims to provide efficient way to align and proportion content
- known as $\mathbf{Flexbox}\ \mathbf{Layout}$
 - idea is to apportion width and height for content
 - proportions relative to container even when their size is unknown or dynamic
- flex layout could, in theory, replace a full grid layout
 - considered more a complement to overall grid structure
- defined flex container expands items to fill the container's available space
 can also shrink them to prevent any possible overflow
- think of a flex layout as supporting multiple directions
 - direction agnostic
- many properties available for **flex**
 - focus upon styling flex container and any flex items

CSS - Flexbox

intro

- helps solve many issues that have continued to plague layout and positioning
- used with HTML elements and components

- both client-side and cross-platform apps
- a few issues it tries to solve
 - vertical and horizontal alignment
 - defining a centred position for child elements relative to their parent
 - equal spacing and proportions for child nodes regardless of available space
 - equal heights and widths for varied content
 - & lots more...

CSS - Flexbox

basic usage

- for any app layout, we need to define specific elements as *flexible boxes*
- i.e. those allowed to use flexbox in a given app
 - e.g.

```
section {
   display: flex;
```

- ruleset will define a section element as a parent flex container
 child elements may now accept flex declarations
- initial declaration, display: flex
 - also includes default values for flexbox layout of child elements
- e.g. <div> elements in a section
 - by default now arranged as equal sized columns with the same initial height

CSS - Flexbox

axes

- elements arranged using flexbox are laid out on two axes
- main axis
 - axis running in the direction of the currently laid out flex items
 - e.g. rows or columns
 - start and end of axis = main start & main end
- cross axis
 - axis running perpendicular to the current main axis
 - start and end of axis = cross start & cross end
- each child element laid out inside flex container called a $\mathit{flex item}$

CSS - Flexbox

flex direction

- set a property for the flex direction
 - defines direction of flex items relative to main axis
 - i.e. layout direction for child elements
- default setting is row
 - direction will be relative to current browser language setting
 - e.g. for English language browsers = left to right





Image - CSS Flexbox





without wrap

Image - CSS Flexbox

spire and the signpost					
Lorem Ipsum Dolor					
Get Distance					
footer tab 1 footer tab 2 footer tab 3 footer tab 4 footer tab 5 footer tab 6					
footer tab 7 footer tab 8 footer tab 9 footer tab 10 footer tab 11					
footer tab 12 footer tab 13 footer tab 14 footer tab 15					

Figure 3: CSS Flexbox - flex wrap

with wrap

Video - Flexbox

flexible design Examples of Modular UI Design Source - Modular UI Design - YouTube

HTML5, CSS, & JS - example - part 5 add flex to grid layout - option 1 • we might specify CSS properties for a flex container

```
.flex-container {
    display: flex; /* defines container as flex */
    flex-direction: row; /* defines positioning of items added to container */
    flex-wrap: wrap; /* defines whether to wrap items to another line */
    justify-content: flex-start; /* defines start point and distribution of items */
```

- allows us to position our container starting at the left
 - items contained in a row
 - contained items wrapping to additional lines if necessary
- many additional options available for each property
- also add rulesets for specific styling of items within a flex container
- we could add properties to a flex item such as
 - specify the order of the flex items
 - whether a particular item can grow or shrink relative to content
 - default size of an item before any remaining space is distributed
 - individual alignment for a given item...

CSS - Flexbox

flex direction reverse

- also set flex direction to reverse
 - starts flex items from the right on an English language browser



Image - CSS Flexbox

spi	re and the signpost
Lo	rem Ipsum Dolor
	Get Distance
	footer tab 6 footer tab 5 footer tab 4 footer tab 3 footer tab 2 footer tab 1
	footer tab 11 footer tab 10 footer tab 9 footer tab 8 footer tab 7
	footer tab 15 footer tab 14 footer tab 13 footer tab 12

Figure 4: CSS Flexbox - flex direction reverse

flex direction reverse

CSS - Flexbox

flex-flow shorthand

- also combine *direction* and *wrap* into a single declaration
 - flex-flow
 - now contain values for both row and wrap

```
– e.g.
```

```
#tabs {
   flex-flow: row wrap;
```

HTML5, CSS, & JS - example - part 6

add flex to grid layout - option 2

• flex container for option 2 design

```
/* note container - flex */
.note-output {
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
    row-gap: 20px;/*applies to rows of items - not above first row... */
    padding-top: 20px;
```

- output notes section
 - organise single notes as flex items
 - add gap between rows of flex items
- justify content in container
 - notes start at left edge, end at right edge
 - space between evenly apportioned per note

CSS - Flexbox

sizing of flex items

- for each flex item, we may need to specify apportioned space in the layout
 - e.g. set space as an equal proportion for each flex item
 - we may add the following to a child item ruleset

```
div.fTab {
   flex: 1;
}
```

- defines each child flex item <div class="fTab">
 - occupy an equal amount of space within the given row
 - after considering margin and padding
- **n.b.** this value is proportional
 - doesn't matter if the value is 1 or 100 &c.
- define additional flex proportions for specific child items

– e.g.

- each odd $\mathit{flex-item}$ will now occupy twice available space
 - space in the current direction

Image - CSS Flexbox

spire and the signpost
Lorem Ipsum Dolor

Get Distance

footer tab 1
footer
tab 2
footer tab 3
footer
tab 4
footer tab 5
footer tab 5
footer tab 7
f

Figure 5: CSS Flexbox - flex item sizing

flex item sizing

CSS - Flexbox

minimum size

• then set a minimum size for a flex item

– e.g.

div.fTab {
 flex: 1 100px;

• or a relative unit for the size

```
div.fTab {
    flex: 1 20%;
}
```

- each flex item will initially be given a minimum
 - e.g. 20% of the available space
 - the remaining space will be defined relative to proportion units

Image - CSS Flexbox

flex item sizing

spire and the signpost

Lorem Ipsum Dolor

	Get Distance	
footer tab 1 footer tab 2	footer tab 3 footer tab 4	footer tab 5
footer tab 6	footer tab 7	

Figure 6: CSS Flexbox - flex item sizing - minimum size

HTML5, CSS, & JS - example - part 7

add flex to notes

- flex container and items useful for organising and positioning our notes
- due to uncertainty about content, size, and general note requirements
 - flex positioning and styling removes the need for assumptions or fixed sizes
- we can start to modify the styling and rendering of our notes using flex

```
/* flex item */
.flex-item {
  flex-basis: 300px; /* default size before extra */
  flex-grow: 1; /* all items will be equal */
```

- gives us a default smallest size for each note
- then the ability for each note to grow to fill the row as required
- also work with responsive layouts
 - due to the minimum size and the option to grow for each item
 - and wrap flex items per flex container
- modify and update styles as we develop travel notes app

DEMO - Travel Notes - grid layout with flex notes

Image - HTML5, CSS, & JS - Flex Notes

Image - HTML5, CSS, & JS - Flex Notes 2

Image - HTML5, CSS, & JS - Flex Notes 3

HTML5, CSS, & JS - example - part 8

add flex to notes

travel notes record notes from various places visited			menu		search	
add note		H	ĸ			
Deterts all						
Leven ipsum delar at annt, consecturar adpission gain. Dones utilizes saturgitati elle consecturar. Dava de saturali elle disa per per consultati. Supportatione expressione en si se consultati. Supportatione expressione en si se consultati en esta esta esta esta esta esta esta interportante esta esta esta esta esta esta esta esta esta esta esta esta esta esta	CANDES			sion .		BORKY
Leven journ doler si auset, consecturar dipiscie gilt. Donce ultrices succipit les consecursos. Casa de succi di si di les esperi- constituit. Supportaise regranta licon mi, si constituit. Supportaise regranta licon mi, si constituit. Supportaise regranta licon mi, si constituit. Supportaise esperatori di la succi internato, licon di la succi di minima di periora dari patricosa. Constituita massi si augest efficitta, es deplos nea frementa, "assate traventi patratti licondaria di la succi magna di la succi di la succi periorato dari patrico di la succi di la succi licondaria di la succi magna di la succi di la succi forgiat, es neccipit magna di estenza. Dones resontanti sinciali farenza di mentenzio.	antibes			juan les pins		Lorem ipsum dolor at anet, consecture adjusting dit. Done thiose subjects subject to come strame. Can subject to adjustication of the common strame can be adjustication of the common in the subject to the subject to the target sequences and the subject to the subject manual adjustication of the subject to the manual adjustication of the subject to the manual adjustication of the subject to manual adjustication of the subject labority. For the subject to the subject labority. For the subject to the subject labority of the subject to the subject labority of the subject to the subject labority of the subject labority. The subject labority of the subject labority of the subject labority of the subject labority of the subject labority of the subject labority of the subject labority of the subject labority of the
***		Lorem ipsum i ultrices suacijo sugue. Suspen Suspendisse ej In varius vel tu pulvinar dui p Donec aliquan fermentam. Pr neo felis hende Suspendisse ul magna elemen	dolor sit amet, or it leo cursus curs disse possere ne gestas lacus mi, argis eget vehicu ulvinar sed. Curs n massa id sapier esent venenatis rerit, cursus leo a llamcorper lacus turn. Donoc vens	mectetur adipiscing elit. Donore un: Cras ao sagittis elit, neo porte etillas equet convalis. es commodo lorem accumanto es las Sed iaculis finibus nulla, eu bibut traisique Intonce suisanot, efficitur, es dapibus nune pharters loboris. Pelentesque ecumana, dignissim nibh, eo saugue fregita, eu suscipit matis laculis fermentum	st tropez	
app's copyright information, additional Tinks						

Figure 7: Grid Layout - flex notes

• Notes with Flex and Media Queries

HTML5, CSS, & JS - example - part 9

add flex to notes - option $\mathbf{2}$

- define styling for flex items in option 2 design
- note defined using card layout design
 - card-view, card-content



- card is flex container for child flex items
 - e.g. note content, header, footer &c.
- **flex** defines shorthand property
 - $-\,$ flex-grow, flex-shrink, flex-basis
 - note set to initial length of $\ \tt 250px$

travel notes record notes from various places visited			
nenu			
earch			
add note add			
innes	nice		
ionaco	menton		
orem ipsum dolor sit amet, consectetur lipiscing elit. Donce ultrices suscipit loo arsus cursus. Cras ac sagittis elit, noc porta ugue. Suspendisse opsuzere nec tellus eget omrado lorem accumsan eu. In varius vel pris eget vehicula. Sed iacuis finibus nulla, a pulvinar dui pulvinar sed. Curabitar istique thoncus euismed. Donce aliquam assa id sapien efficitar, eu dapibus nunc immentum. Praesent venenatis phartera bortis. Pellentesque nec felis benderit, arsus los accumsan, dignissim nibh. uspendisse ultamcorper lacus eu augue ragiat, eu suscipit magna elementam. Donce enenatis iaculis fermentum	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donce ultrices suscipit leo cursus cursus. Cras ac sagitti selit, neo porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse goptas lacus mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis finibos nulla eu pulvinar dui pulvinar sed. Curabitur tristique thoncus euismod. Donce aliquam massa id sapien efficitur, eu dapibus nunc fermentam. Praesent venenatis pharetra lobottis. Pellentesque nec felis henderit, cursus leo accumsan, dignissim nibh. Suspendiseu ultancorper lacus eu augue feugiat, eu suscipit magna elementam. Done venenatis iaculis fermentum		
ntibes	st tropez		

Figure 8: Grid Layout - flex notes - medium

record	notes from various places visited
menu	
search	
add not	te
	add
Delete all Lorem ipsu Cras ac sag igestas lacu finibus null	m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus ittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse s mi, ac commodo lorem accumsan eu. In varius vel turgis eget vehicula. Sed iaculis a, eu pulvinar dui pulvinar sed. Curabitar tristique rhoncus euismod. Donec aliquam
Delete all Lorem ipsu Cras ac sag rgestas lacu finibus null massa id sa Pellentesqu acus eu au	m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus ittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse as mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis a, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam pien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. e nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper gue feugiat, eu suscipit magna elementum. Donec venenatis isforterit
Delete all Lorem ipsu Cras ac sag gestas lacc finibus null massa id sa Pellentesqu acus eu auj cannes	m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus ittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse as mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis a, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam pien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. e nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamoorper gue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum
Delete all Lorem ipsu Cras as sag egestas lacc finibus mull massa id sa Pellentesqu acus eu au annes	m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus ittis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse as mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sed iaculis a, eu quevinar dui pulvinar sed. Curabitar tristique rhoneus eusismod. Donec aliquam pien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. e nec felis hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper gue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum
Detete all Lorem ipsu Cras ac sag rgestas lact finibus null massa id sa Pellentesqu acus eu au cannes monaco Lorem ipsu Cras ac sag rgestas lact finibus null massa id sa Pellentesqu acus eu au	m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus titis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse as mi, ac commodo lorem accumsane eu. In varius vel turpis eget vehicula. Sed iaculis a, cu pulvinar dui pulvinar sed. Curabitur tristique rhoncus euismod. Donec aliquam pien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. e nec feils hendrerit, cursus leo accumsan, dignissim nibh. Suspendisse ullamcorper gue feugiat, eu suscipit magna elementum. Donec venenatis iaculis fermentum m dolor sit amet, consectetur adipiscing elit. Donec ultrices suscipit leo cursus cursus titis elit, nec porta augue. Suspendisse posuere nec tellus eget convallis. Suspendisse as mi, ac commodo lorem accumsan eu. In varius vel turpis eget vehicula. Sedi aculis a, eu pulvinar dui pulvinar sed. Curabitur tristique rhoncus eusismd. Donec aliquam pien efficitur, eu dapibus nunc fermentum. Praesent venenatis pharetra lobortis. e nec feils hendrerit, cursus leo accumsan, dignissim nibb. Suspendisse ullamcorper gue feugiat, eu suscipit magna elementum. Praesent venenatis pharetra lobortis.

Figure 9: Grid Layout - flex notes - small

CSS - Flexbox

flex item alignment

- Flexbox allows us to define alignment for flex items in each flex container
 - relative to the main and cross axes
- e.g. we might want to specify a centred alignment for flex items

```
#tabs {
   flex-direction: row;
   flex-wrap: wrap;
   align-items: center;
```

- align-items: center
 - causes flex item in flex container to be centred along the cross axis
 - however, they'll still maintain their basic dimensions
- also modify value for align-items to either flex-start or flex-end
- such values will align flex items to either start or end of cross axis

CSS - Flexbox

override align per flex item

- as with flex
 - also override alignment per flex item
 - using align-self property add a value for positioning
- e.g. a sample declaration might be as follows

```
div.fTab:nth-child(even) {
  flex: 2;
  align-self: flex-end;
}
```

CSS - Flexbox

justify content for flex item

- also specify justify-content for flex items in a flex container
- property allows us to define position of a flex item relative to main axis
- default value is flex-start
- then modify it relative to one of the following
 - flex-end

```
- center
```

- space-around

* distributes each flex item evenly along main axis with space at either end

- space-between
 - * same as space-around without space at either end...

CSS - Flexbox

alignment and order - part 1

- define alignment relative to each axis using a specific declaration
 - e.g. for the main we may use justify-content
 - for the cross axis we use align-items
- also modify layout order of flex items
- without directly changing underlying source order
- use the following pattern to specify order

<pre>div.fTab:first-child {</pre>		
order: 1;		
}		

• first flex item will now move to the end of the tab list

Image - CSS Flexbox

spire and the sig	gnpost					
Lorem Ipsum D	olor					
		0-1	Distance			
		Ge	Distance			
footer tab 2	footer tab 3	footer tab 4	footer tab 5	footer tab 6	footer tab 7	footer tab 1

Figure 10: CSS Flexbox - flex item order 1

flex item order

CSS - Flexbox

alignment and order - part 2

- due to default order for flex items
 - by default, all flex items have an order value set to 0
- higher the order value, later the item will appear in the list &c.
- items with the same order will revert to the order in the source code
- also possible to ensure certain items will always appear first
 - or at least before default order values
 - by using a negative value for the order declaration
 - e.g.

div.fTab:last-child {
 order: -1;
}

CSS - Flexbox

nesting flex containers and items - part 1

- Flexbox can also be used to create nested patterns and structures
 - e.g. we may set a flex item as a flex container for its child nodes
- we might add a banner to the top of a page



CSS - Flexbox

nesting flex containers and items - part 2





• we get a simple proportional split of 4:1 for the input field to the button

Image - CSS Flexbox

nested flex containers

spire and the signpost point to the stars			Search
	Get Distance		
footer tab 2 foo	ter tab footer tab 4	footer tab 5 footer tab 6	footer tab

Figure 11: CSS Flexbox - nested flex containers

HTML5, CSS, & JS - example - part 10

add flex to notes - option 2

- define rulesets for child items
 - card-view header
 - card-view footer

```
.card-view header {
    padding: 10px;
    background-color: #6666666;
    color: #EEEEEE;
    font-size: 17px;
}
.card-view footer {
    border-top: 1px solid #6666666;
    padding: 10px 0;
}
```

• DEMO - Travel Notes - Version 3 - Grid

Image - HTML5, CSS, & JS - Flex Notes

Image - HTML5, CSS, & JS - Flex Notes

CSS grid layout - part 8

media queries

- often need to consider a mobile-first approach
- introduction of CSS3, we can now add media queries
- modify specified rulesets relative to a given condition
- eg: screen size for a desktop, tablet, and phone device
- media queries allow us to specify a breakpoint in the width of the viewport

 will then trigger a different style for our application
- could be a simple change in styles
- such as colour, font etc
- could be a modification in the grid layout

ard	head	or
aru	neau	CI

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

card footer...

Figure 12: Grid Layout - flex notes - card design

add note	266	image search	search
note controls			
card header	Lavers spans data tak senet, converterator allopicity att, bonec, para tak senet and allow conver- porter tradition prime are significant production prime are significant production prime are significant production prime and the senet response to the senet and the senet response to the senet and the senet many and texts. Self of entities the senet and the senet are senet and the senet and the senet and the senet and the senet and the senet prime and the senet and the senet many and texts. Self of entities the senet and the senet and the senet prime and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet text and the senet and the senet and the senet and the senet and the senet text and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the senet and the sen	Leven loans dobr is enet, consected a sign is bob block. The second second second second second consected second second second consected second second second provide second second second second pathors, imperiet level on workstate needs, bit solarity and second second second second second second second second second second second second second second second second second pathors is an experience of second second second second second second	Lorent losant dolor sit and; placent support an Local Stehendt a solit-Carlin quant interdam, placent support and the solitation between the solitation of the solitation placent interdam solitation with a compare stands. Examination were unreased texture and a level strack compare stands. Examination were placent and the solitation of the compare stands. Examination were placent of solitation and a level placent of solitation and a level placent of solitation and a level placent of solitation and a level solitation at versit.

Figure 13: Grid Layout - flex notes - card view with space between

- effective widths for our columns per screen size etc...

Image - Grid Layout 4



Figure 14: Grid Layout - Media Queries

CSS3 Grid - responsive layout

intro

- display a layout with a variety of patterns and structures, e.g.
 - single column for a phone
 - add a sidebar for a tablet of lower window resolution
 - $-\,$ full width view with dual sidebars &c.
- use responsive designs and structures for various games, media playback...
- responsive works with variety of markup
 - e.g. transform SVG designs

CSS3 Grid - responsive layout

page structure

• start with a sample page structure for a HTML page

CSS3 Grid - responsive layout

```
page structure - HTML5
```

```
• add some HTML5 markup for a header , navigation , footer , and some main content
<div class="wrapper">
   <header class="site-header">
       <h3>Spire & the Signpost</h3>
       <h5>Shine through the gloom, and point to the stars...</h5>
   </header>
    <nav class="site-nav">
       <a href="">Home</a>
           <a href="">Charts</a>
           <a href="">Data</a>
           <a href="">Views</a>
       </nav>
   <aside class="content-side">
       <header>
           <h5>sidebar...</h5>
       </header>
   </aside>
    <main class="content">
       <article class="content-article">
           <header class="article-header">
               <h5>Welcome</h5>
           </header>
           ...
       </article>
   </main>
```

```
<section class="site-links">
        <h6>social links...</h6>
   </section>
   <footer class="site-footer">
        <h6>footer...</h6>
    </footer>
</div>
```

• demo - basic responsive

CSS3 Grid - responsive layout

CSS and structure - part 1

- for the page structure
 - need to define some template areas for our grid in the CSS

– e.g.

```
.content {
   grid-area: content;
```

- use such template area names
 - defined with the grid-area property
 - define a layout for the overall page or part of a page

CSS3 Grid - responsive layout

```
CSS and structure - part 2
```

• template areas may then be used with the parent for the grid structure - e.g. wrapper for the overall page

٠

- wrapper class will display as a grid
 - with a gap between each area of the grid
 - has a single column in this example
 - includes the required order for the grid areas

CSS3 Grid - responsive layout

define media query

- current example would be suitable for a collapsed phone view
 - single column view
 - will also render for other resolutions and devices
- then add a media query for alternative layouts and displays
 - may be triggered using a check of current width for screen
 - check width of window...

```
/* min 700 */
@media (min-width: 700px) {
    .wrapper {
        grid-template-columns: 1fr 3fr;
        grid-template-areas:
            "site-header site-header"
            "site-nav site-nav"
            "content-side content"
            "site-links site-footer"
        }
}
```

CSS3 Grid - responsive layout

specific media query

- add further media queries to handle various rendering requirements
 - e.g. add height property to fix footer at bottom of page

```
@media (min-width: 700px) {
    .wrapper {
      grid-template-columns: 1fr 3fr;
      grid-template-rows: 120px 60px calc(98vh - 240) 60px;
      grid-template-areas:
        "site-header site-header"
        "site-header site-header"
        "site-nav site-nav"
        "content-side content"
        "site-links site-footer";
      height: 98vh;
    }
}
```

- specify height of current *viewport* using a relative unit, vh
- add grid-template-rows to define known heights for three of the four rows
- add a variant height for the main content
 - main content is only given a height corresponding to available space in viewer window
 - height achieved using the calc() function
- demo responsive with specific media query

HTML5, CSS, & JS - example - part 11

add responsive design - option $\mathbf{2}$

• add new CSS - responsive.css

• update index.html with new <link>

```
<link rel="stylesheet" href="assets/styles/responsive.css">
```

- define initial media queries, 915px and 545px
- 915px for desktop to tablet
- 545px for tablet to phone &c.

HTML5, CSS, & JS - example - part 12

add responsive design - option 2

- media query for 915px
 - update wrapper, banner
 - adjust site-header & banner extras

div.wrapper {

```
grid-template-rows: 80px auto 80px;
margin: 20px 10px 0 10px;
}
div.banner {
  grid-template-rows: 80px;
  grid-template-areas:
       "site-logo site-header"
}
.site-header {
  margin-right: 0;
}
.banner-extras {
  display: none;
}
```

- need to update note controls, main content
- e.g. update flex for note card design
 better use of available width
- _____

Image - HTML5, CSS, & JS - Media Query - $915\mathrm{px}$

HTML5, CSS, & JS - example - part 13

add responsive design - option 2

- update note controls
 - move to stacked rows
 - one row per option, input field

```
.page-heading {
   grid-template-columns: 100%;
   grid-template-areas:
        "add-note"
        "search-images"
        "note-controls";
}
```

travel notes	
add note	image search search
note controls Delete all	
card header	Lorem ipsum dolor sit amet, consectetur adipiccing elit. Donec placerat sapien ac lacus bibendum, a solicitudin quam interdum. Donec tristique purus nec sapien tempor rutum. Proin eget quam pulvinar, imperdiet felis eu, vestbulum sem. Ut pellentesque venenaits neque, la volutata dina corgue si a meet, im, mi bictusara corgue si anet, in, mi bictusara corgue si anet, in, mi bictusara portitor diam. In et arcu rhoncus, tinciduro do a, interdum est. Donec eu nisi totror. Nulla vel volutata netus. Sed justo erat, scelerisque et prefum non, solicitudin et velit.
Lorem ipsum dolor sit amet,	Lorem ipsum dolor sit amet,

update HTML for input group

 container for input field and button

```
<div class="input-group">
    <input type="text" id="input-note" />
    <button id="add-note">add</button>
</div>
```

• update CSS with flex for input group

```
.input-group {
    display: flex;
    justify-content: space-around;
    flex-grow: 1;
```

Image - HTML5, CSS, & JS - Media Query - 915px

HTML5, CSS, & JS - example - part 14

add responsive design - option $\mathbf 2$

- update input group with modified styles, aesthetics, same sizes &c.
- add ruleset for input-group to style.css only
 style applied regardless of page width

/* group input and button *
.input-group {

\square	travel notes	783px × 955p
add no	te	add
image	search	search
note c	ontrols	

Figure 16: Media Query - 915px - note controls & input

Image - HTML5, CSS, & JS - Media Query - 915px

Figure 17: Media Query - 915px - input group - desktop

References

- MDN CSS Flexbox
- W3 Schools CSS Flexbox
- Various
 - Example Responsive UI Designs YouTubeMDN CSS3 Grid

 - Modular UI Design YouTube