



НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ



Responsive Design

The viewport

- Буквално: "изглед" или начин по-който ще се вижда страницата на екрана
- Отнася се до размера, който се приема за размер на екрана
- При настолните и преносимите компютри този размер е винаги толкова, колкото е голям прозореца на браузъра
- при мобилните устройства като таблет и телефони не е така
- те имат фабрично вградена симулация на по-голям екран, за да могат да показват целите уеб страници

480px
(screen resolution)

800px
(screen resolution)



This page has a viewport width of 980 pixels. This is how it looks at 100% zoom on an HTC Desire.

480px
(screen resolution)

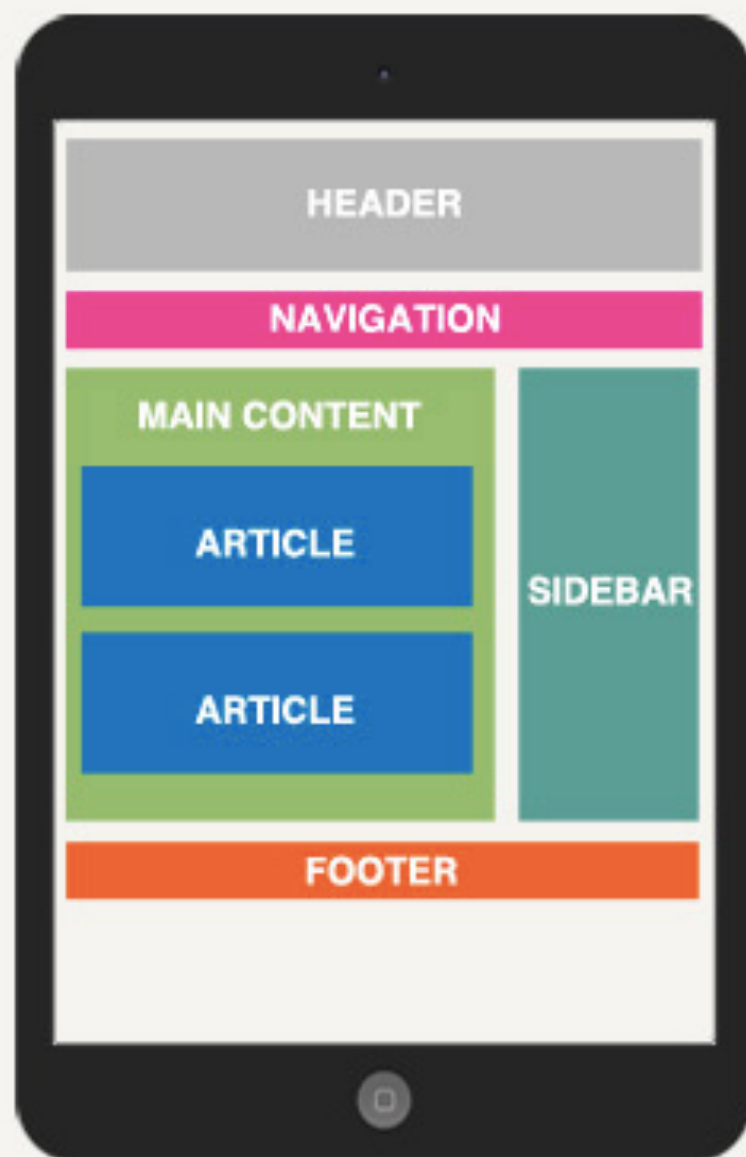
800px
(screen resolution)



The viewport is still 980px wide, but since the page is zoomed out it is squeezed into 480px, the resolution of the phone's screen.

meta name="viewport"

- този мета таг "казва" на браузъра да зададе определена ширина на viewport-a
- ако мета тагът липсва, браузърът се държи като на голям екран и на малки устройства показва страницата смалена (за да се вмести в екрана)
- по този начин @media правилата (media queries), които сме описали в CSS-a никога не се изпълняват
- правилният начин да използваме този мета таг е следният:
<meta name="viewport" content="width=device-width, initial-scale=1">



Without the viewport metatag

Device renders website with a width of 1280px



With the viewport metatag

Device renders website with a width of 320px

CSS3 @viewport

- Общо взето прави същото като мета тага, но е в css файла
- аз лично препоръчвам да се използва мета тагът и само ако има нужда от допълнителни настройки - да се задава @viewport в CSS-a
- <https://developer.mozilla.org/en-US/docs/Web/CSS/@viewport>

Viewport 1-minute-long lessons

- <https://www.youtube.com/watch?v=6P9uLyvcd3Y>
- <https://www.youtube.com/watch?v=kqHKOSd5NCk>
- <https://www.youtube.com/watch?v=UEZYIDb7Tt4>
- <https://www.youtube.com/watch?v=F-akKxve2oE>
- <https://www.youtube.com/watch?v=sFzdU8Z1Wd8>

основни CSS правила за responsive design

- **max-width: 100%** на блоковите елементи, ще им попречи да "излизат" извън екрана
- `box-sizing: border-box`
- `font-size: 1.5em`
- бутоните не трябва да стават по-дребни от колкото е върхът на показалеца. За целта използваме `min-width` и `min-height`

media queries

Как се ползват

- @media правилата са нови за CSS3 и имат малко по-странен синтаксис

- Синтаксис:

```
@media screen and (max-width: 320px) {  
    /* CSS styles come here */  
}
```

- или директно в head-а на html документа:

```
<link rel="stylesheet" media="(max-width: 800px)" href="example.css">
```

- примери:

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

https://www.youtube.com/watch?v=g1Pu_qQs8t0



ONE DOES NOT SIMPLY

WRITE A CSS3 MEDIA QUIRY

@media query 1-minute-long lessons

<https://www.youtube.com/watch?v=AmHW9Hhm5ml>

<https://www.youtube.com/watch?v=xvBbwVQqDWE>

https://www.youtube.com/watch?v=KtqM_xH-pLY

https://www.youtube.com/watch?v=g1Pu_qQs8t0

<https://www.youtube.com/watch?v=gCiCojJGr9w>

<https://www.youtube.com/watch?v=17XgkPFq6eY>

<https://www.youtube.com/watch?v=17XgkPFq6eY>

Breakpoints

- и.к: "точки на пречупване"
- когато пишем @media queries, трябва да си изберем точки на "пречупване", където стиловете ще се сменят
- тези breakpoints, казват при каква ширина на екрана ще се изпълни дадено @media правило
- Mobile first! - това е популярна стратегия, при която се изгражда първо мобилната версия на сайта, след което се преминава към следващия по големина device и т.н. - до най-големия

The grid fluid system

- това е комбинация от `display: table, float: left` и `width`
- елементите, на които се прилагат грид класовете, се натрупват един след друг хоризонтално, докато не запълнят цялото пространство и тогава продължават на долния ред
- работи добре благодарение на `display: table-row` пропъртито, което ни спестява да пишем `clear: both`
- <https://www.youtube.com/watch?v=0uUV5aZfkIA>

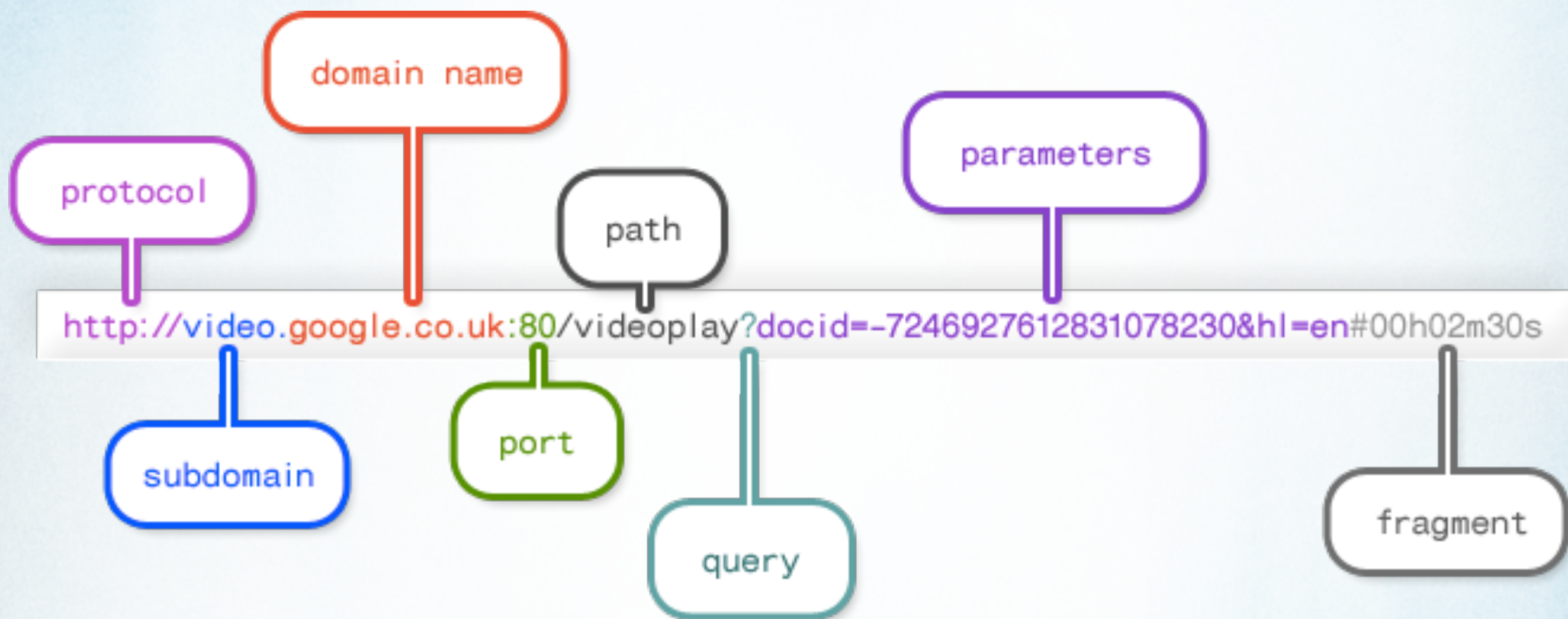
въпроси?

URLs



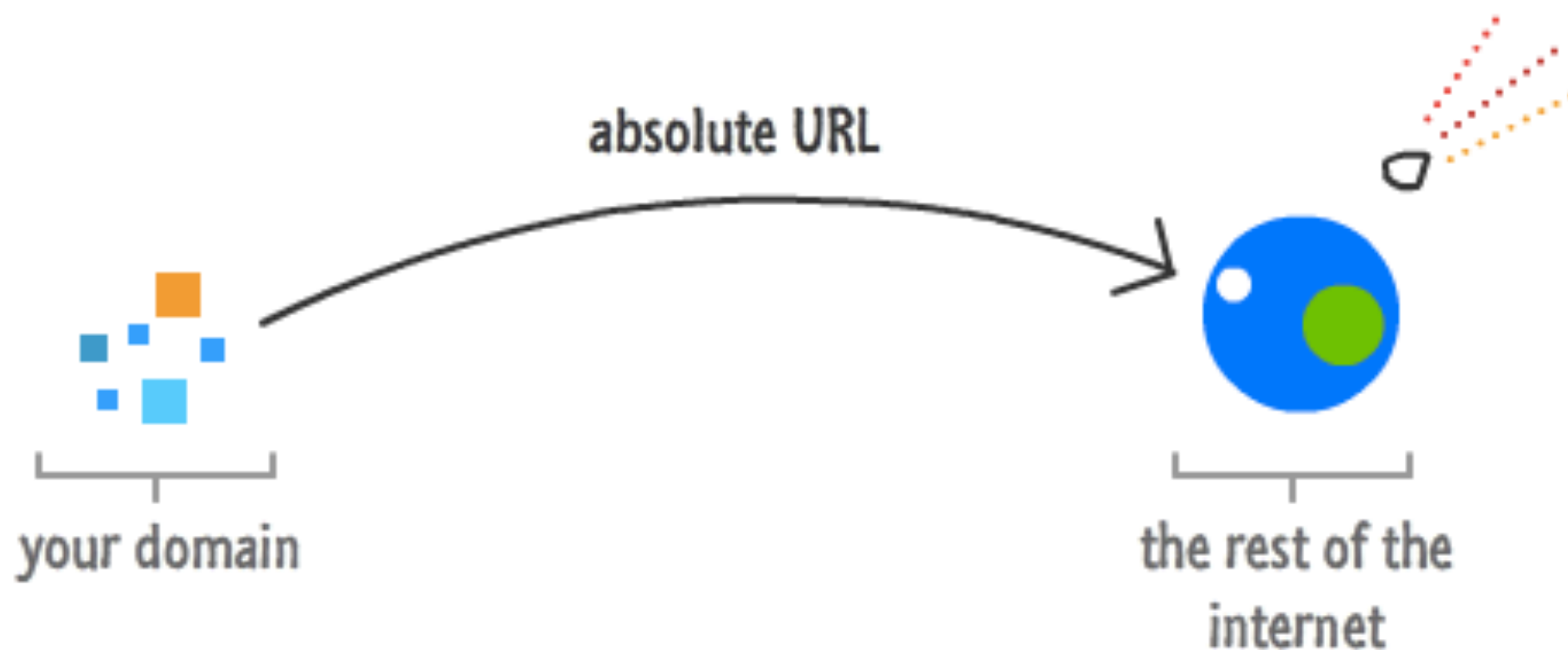
Основни URL компоненти

- протокол
- домейн
- път до ресурс
- search query



URLs and paths

- Когато link-ваме ресурс, който се намира в същия проект (уебсайт), използваме само пътя до ресурса
- Когато ресурса е на друг сървър или принадлежи на друг проект (уебсайт) - използваме url



Относителен и пълен път

- Има два начина да локализираме ресурс (файл), който се намира в същия проект: чрез относителен и чрез абсолютен (пълен) път до файла
- Относителен път до ресурса е локацията на този ресурс, спрямо локацията на текущия файл
- Абсолютен път до ресурс е пътя до ресурса спрямо root-а на проекта (уебсайта)
- Абсолютният (пълният) път до ресурс, съвпада с пътя до ресурса в url записа

Относителен и пълен път във файловата система

Our code inside style.css looks like this:

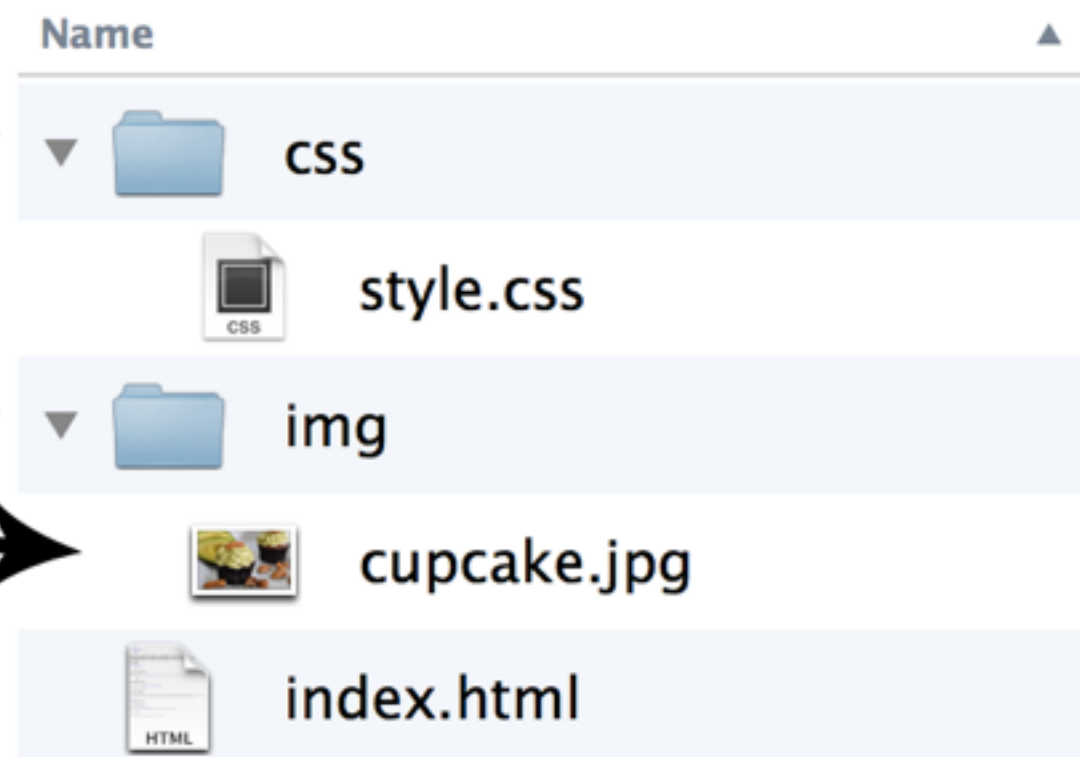
```
background-image: url('../img/cupcake.jpg');
```

1. We start in style.css inside the css folder.

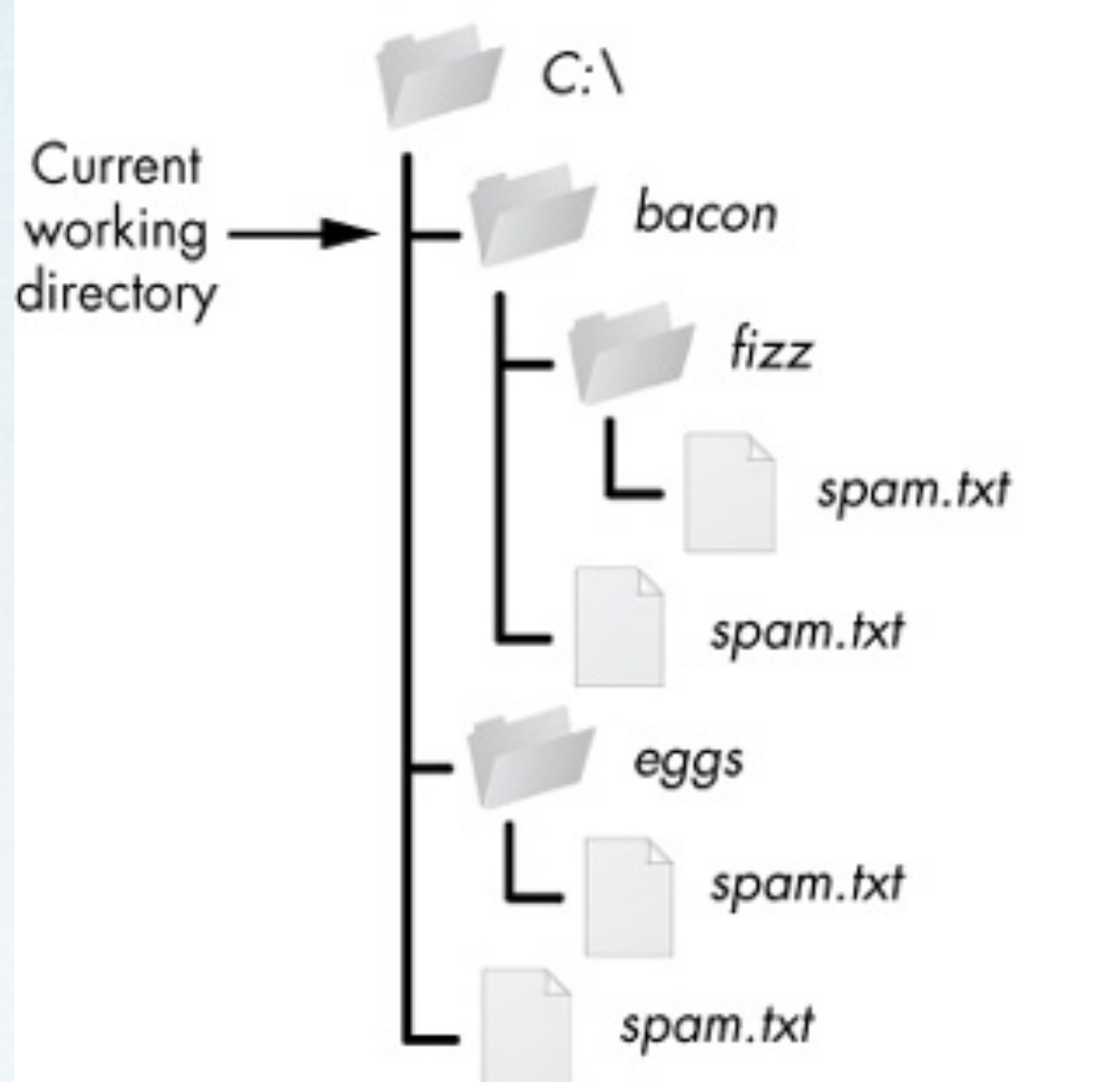
2. The “..” goes *up* one directory.

3. The “/img” goes *down* into the img directory.

4. The “/cupcake.jpg” gets the image.



Относителен и пълен път във файловата система

	Relative Paths	Absolute Paths
	..\	C:\
	.\	C:\bacon
	.\fizz	C:\bacon\fizz
	.\fizz\spam.txt	C:\bacon\fizz\spam.txt
	.\spam.txt	C:\bacon\spam.txt
	..\eggs	C:\eggs
	..\eggs\spam.txt	C:\eggs\spam.txt
	..\spam.txt	C:\spam.txt

Bootstrap

- готова обвивка (framework) за фронт-енд решения
- с него получаваме наготово:
 - css reset
 - css база за изграждане на responsive design (grid система, цветове, фонтове и т.н.)
 - предварително изготвени компоненти (като формуляри, диалогови прозорци, навигационни панели и т.н.)
 - javascript функционалност
- <http://getbootstrap.com/>

Домашно

<https://github.com/zzeni/swift-academy-homeworks/blob/fe-03/tasks/L06>