The Mobile Web

Web Engineering

Prof. D. König
Christian Ribeaud

Responsive Design

Respond to varying screen sizes/resolutions

Prerequisite Knowledge:

HTML, CSS, JavaScript, Web-MVC

Testing the WebMile

The variety of devices, screen sizes, and resolutions makes any testing difficult.

Provider and products change quickly: google, microsoft, saucelabs, mobiready, …

Approaches

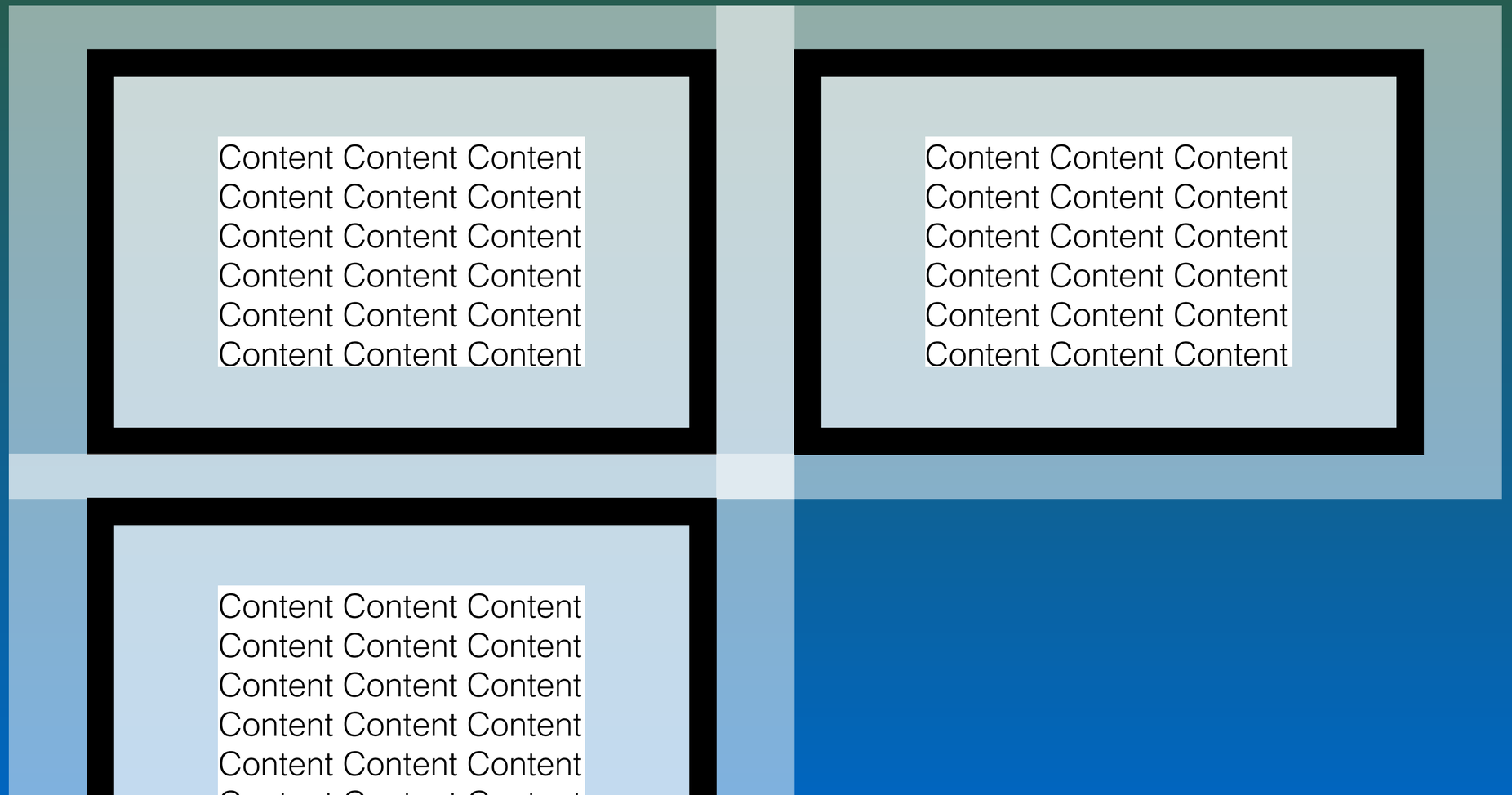
Flexible layout (CSS)

Media queries (CSS)

Dynamic in-page logic (HTML, JS)

Serve different views per capability (MVC)

Flexible Layout { float: left; }



Media Query (CSS)

```
#title { width: 50%; }

@media screen and (max-width: 800px) {
   #title { width: 100%; }
}
```

Media Query (CSS)

```
<link rel="stylesheet" type="text/css" media="screen and (min-width: 361px) and (max-width: 480px)" href="landscape.css">
```

Media Query Attributes

max-width, max-device-width,
min-width, min-device-width, *(height)*
orientation (portrait, landscape),
[min-,max-,device-]aspect-ratio,
color, resolution, touch-enabled, …

Dynamic in-page logic

```
<body onresize="adapt()">


<script>
   function adapt() { …; screen.size … }
</script>
```

screen vs. window vs. page

View per Capability

How to detect the capability?

How to change the view?

Request - Response

server

controller

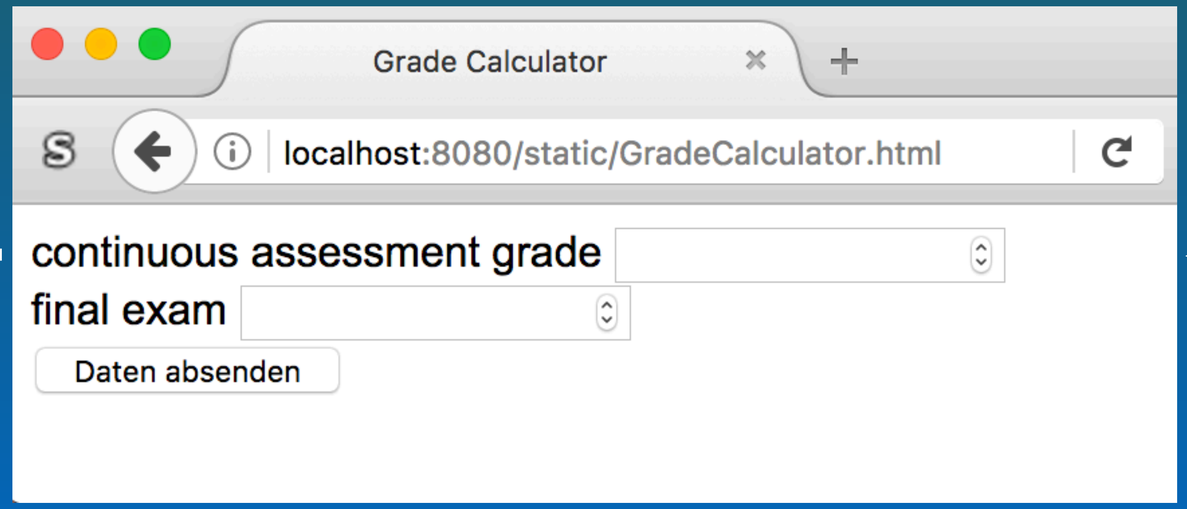
```
class MyController
  def myAction(en, exam)
```

request

response

client

view



How to change the View?

Use different CSS

Select a different layout

Rendering a different view

When to use: rule of thumb

CSS float	*Always consider*
@media	*Mostly static content*
onresize	*Fine-grained control*
MVC	*Default*