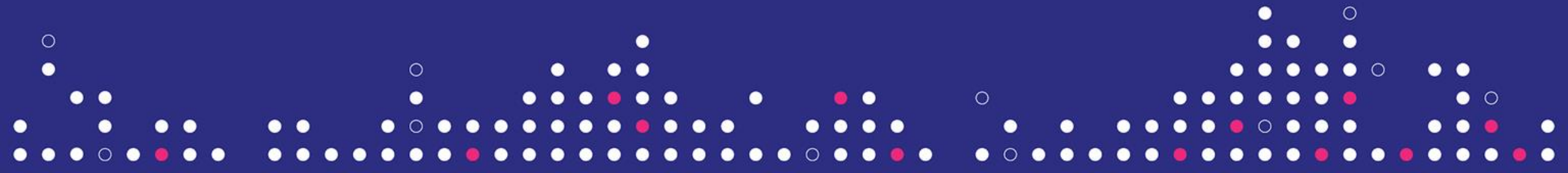


# BOOTSTRAP 4

Luděk Roleček, Petr „Trohat“ Sedláček



Co budeme dneska dělat



1. trochu opakovat

2. povídat si o knihovnách

3. nahlížet pod pokličku

4. taky vytvoříme moc hezkou stránku

# RESPONZIVNÍ WEBDESIGN

## ZKRÁCENĚ RWD

# Co je RWD?



Cílem je:

**zajistit optimální uživatelskou zkušenost na  
mnoha různých zařízeních**

(obsah, navigace, ...)

MOBILY VYHRÁVAJÍ

# Mobily vyhrávají



Neustále roste podíl mobilů na:

- Počtu navštívených stránek
- Času stráveném na webu

Dle různých zdrojů se dnes uvádí i více než 50% podíl.

Je to stále rostoucí tendence.

Stránka se přizpůsobuje



## **Upravuje svoje rozložení a obsah podle prostředí, ve kterém je prohlížena.**

Nejčastěji přizpůsobujeme podle velikosti dostupné zobrazovací plochy, ale lze i podle mnoha dalších kritérií (landscape, color/monochrome, print, tv, handheld, atd.).

## **Mobile first přístup:**

- Výchozí je rozložení pro malá mobilní zařízení.
- Progresivně vylepšujeme pro tablety a počítače (a pro dostupné podporované technologie).



# ZÁKLADNÍ TECHNIKY RWD

1.flexibilní obrázky

2.CSS Media Query

3.responzivní grid

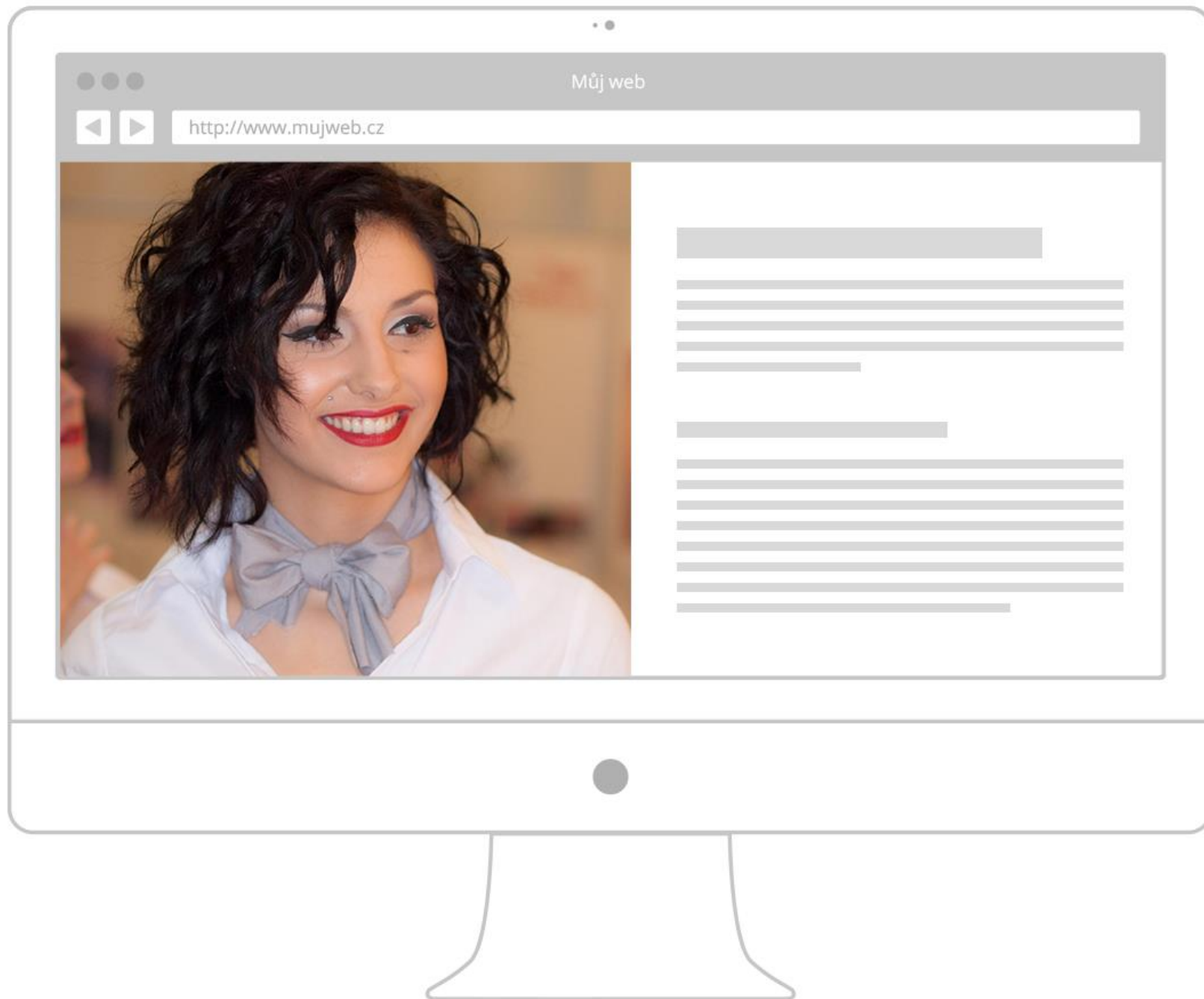
# 1. FLEXIBILNÍ OBRÁZKY

**Velký obrázek na malém zařízení = problém.**

**Vše se musí přizpůsobovat.**

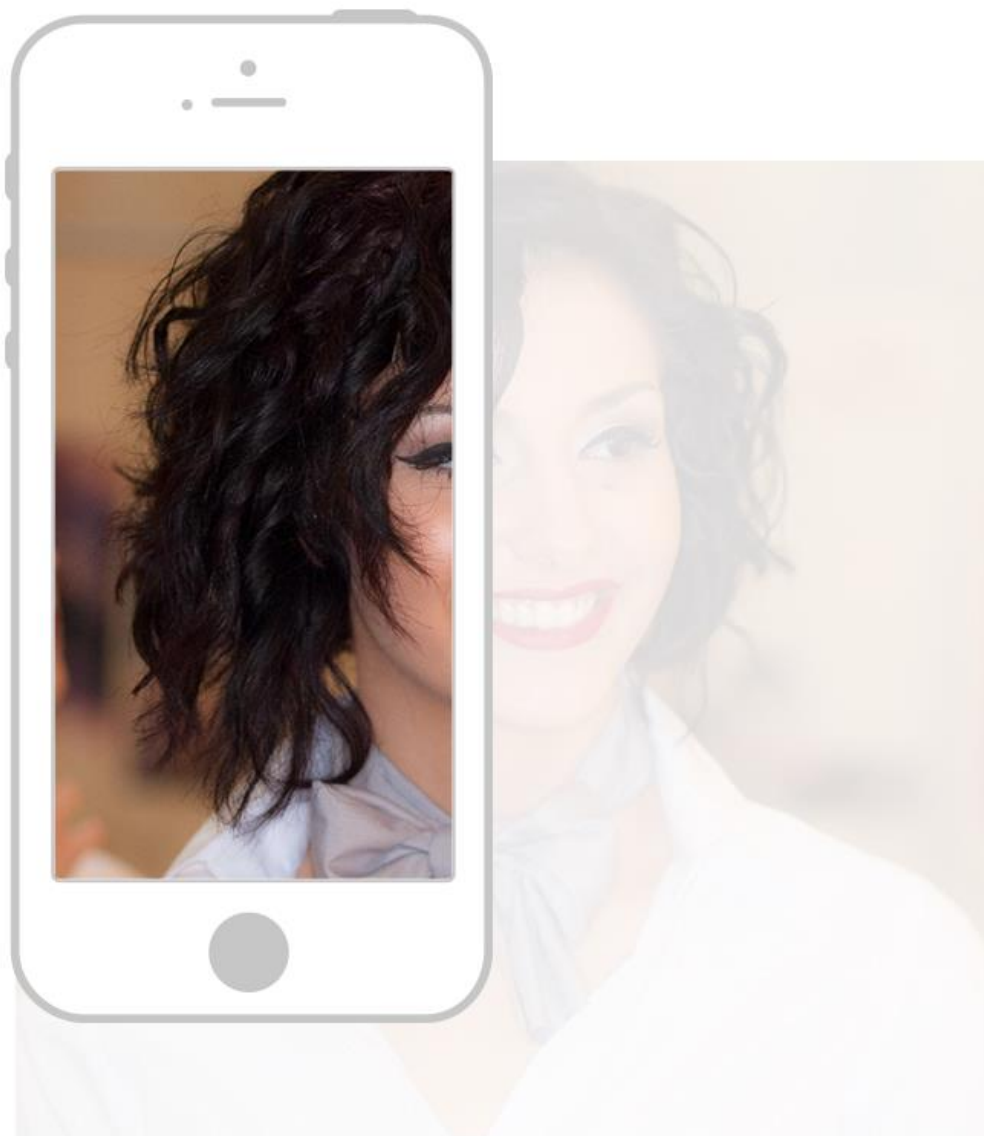
Ideálně načítáme různé obrázky pro různá rozlišení/zařízení.

My dnes budeme kvůli zjednodušení pracovat pouze s jedním obrázkem pro všechny breakpointy.

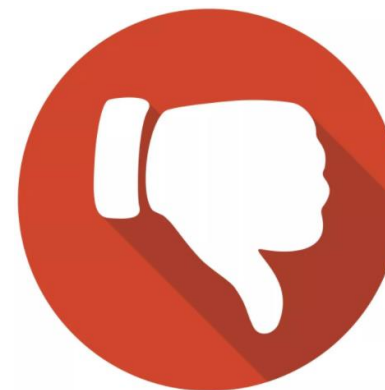


**Velký obrázek  
na PC**





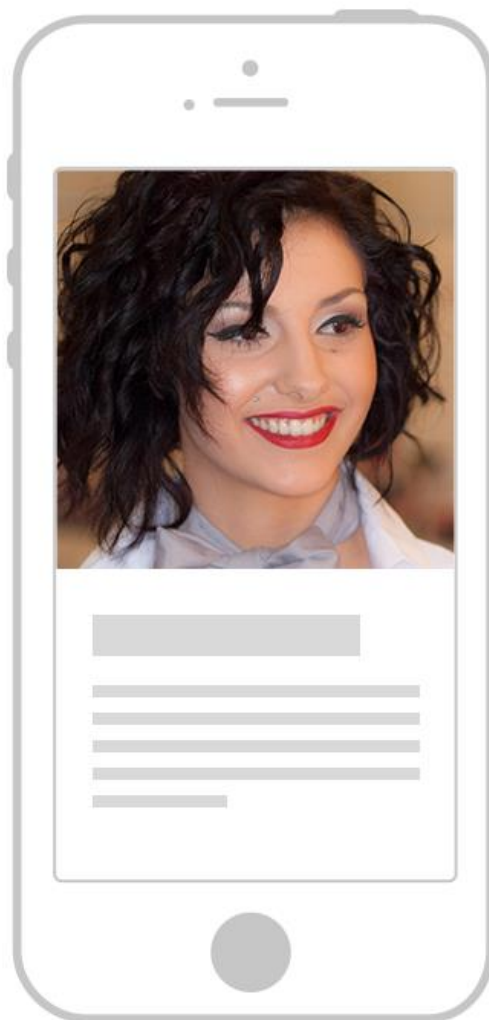
**Velký obrázek  
na mobilu**





**Responzivní  
(flexibilní / fluidní)  
obrázek  
na mobilu**





## **Responzivní (art direction) obrázek na mobilu**





# Flexibilní obrázky v CSS

CSS

```
img {  
    width: 100%;  
}
```

Lze použít i **max-width**, záleží na záměru autora stránky.

## 2. CSS MEDIA QUERY

**Nejčastěji reagujeme na šířku okna prohlížeče / zařízení.**

(ale existuje mnoho dalších vlastností)

## Technika 2 - CSS Media Query

CSS

```
@media (min-width: 800px) {  
    /* css styly */  
}
```

Nejčastěji: **min-width** nebo **max-width**, lze kombinovat pomocí **and**

# Breakpoint



- Hodnotě uvnitř v media query říkáme **breakpoint**
- Doslova místo, kde se “láme” layout (stránka mění své rozložení)

# RESPONZIVNÍ GRID

## "MŘÍŽKA"

## Technika 3 - Responzivní grid



- bloky na stránce skládáme na tuto mřížku
- umožňuje rychle tvořit variabilní layouty
- lze definovat různá rozložení pro různá zařízení
- napsat vlastní řešení vyžaduje dobrou znalost CSS, a proto...

# CSS FRAMEWORK



# CSS framework



- sada CSS stylů, která usnadňuje tvorbu webu
- stačí použít, nemusíme psát (moc) vlastní CSS
- nemusíme “umět CSS” (diskutabilní)
- základem skoro každého CSS frameworku je grid

# CSS frameworky



- **Bootstrap**
- **Foundation**
- Bulma
- Semantic UI
- a spooooooooouuuusta dalších...

BOOTSTRAP 4

# [getbootstrap.com](https://getbootstrap.com)

- stažení, dokumentace, příklady
- dokumentaci otevřít do záložky a nezavírat, budeme potřebovat celý den :)

# Tři věci, co budeme brát:

1. Grid systém
2. Utility třídy
3. Komponenty

## Tři pravidla pro Bootstrap:

1. Bootstrap je o přidávání tříd
2. Bootstrap je jenom o přidávání tříd
3. Bootstrap je (skoro vždycky) jenom o přidávání tříd

# 1. RESPONZIVNÍ GRID

## "MŘÍŽKA"

# Základní struktura

HTML

```
<container>
```

```
  <row>
```

```
    <col>
```

```
    <col>
```

```
    . . .
```

```
  </row>
```

```
</container>
```



# Container

```
<div class="container"> </div>
```

- obaluje celý grid
- každý řádek musí být uvnitř kontejneru
- dva typy

## **container**

- zabírá pouze určitou šířku stránky (nastavenou Bootstrapem), vlevo a vpravo jsou mezery od okraje stránky

## **container-fluid**

- je od kraje do kraje přes celou šířku stránky

## Řádek - row

```
<div class="row"> </div>
```

- řádek, horizontální sekce na stránce
- vždy uvnitř .container nebo .container-fluid
- obaluje sloupce
- “Není vidět”

## Sloupce - col

```
<div class="col"> </div>
```

- Sloupec, vertikální sekce na stránce
- Rozdělí se rovnoměrně uvnitř řádku
- Nebo lze nastavit na konkrétní šířku v konkrétním breakpointu

```
<div class="col-7"> </div>
```

- Sloupec, vertikální sekce na stránce
- Rozdělí se rovnoměrně uvnitř řádku
- Nebo lze nastavit na konkrétní šířku v konkrétním breakpointu

# Určení šířky sloupce pro breakpoint



`<div class="col-XX-YY"> </div>`

Za XX dosadíme název breakpointu:

- (xs) - extra small, mobil - "xs" se neuvádí, jen col-YY
- sm - small, tablet
- md - medium, desktop
- lg - large, desktop+
- xl - extra large, TV, projektor

# Určení šířky sloupce pro breakpoint



```
<div class="col-XX-YY"> </div>
```

Za YY dosadíme šířku sloupce 1-12:

- 6 = polovina stránky
- 4 = třetina stránky
- 3 = čtvrtina stránky
- atd.



## Určení šířky sloupce pro breakpoint



```
<div class="col-6"> </div>
```

- Na mobilu (xs) bude sloupec zabírat polovinu šířky dostupného prostoru (6 sloupců z 12)

## Určení šířky sloupce pro breakpoint



```
<div class="col-sm-4"> </div>
```

- Na tabletu (sm) bude sloupec zabírat třetinu šířky dostupného prostoru (4/12)

## Určení šířky sloupce pro breakpoint



```
<div class="col-md-3"> </div>
```

- Na počítači (md) bude sloupec zabírat čtvrtinu šířky dostupného prostoru (3/12)

# ÚKOL

**Nastav 2 a 3 sloupce vedle sebe podle zadání v HTML souboru.**

Nezapomeň, že sloupce (**col**) musí být uvnitř řádku (**row**) a ten uvnitř kontejneru (**container** *nebo container-fluid*).

Otevři v editoru složku **Příklady**.

MOBILE FIRST

# Mobile first přístup



- Sloupce bez určeného breakpointu platí pro všechno
- Hodnoty se “dědí” směrem nahoru
- Sloupce bez definované šířky se rozdělí rovnoměrně uvnitř řádku
- Stacking columns - sloupce s šířkou 12 jsou pod sebou

# Variabilní sloupce



- Šířka sloupce může být variabilní
- Sloupec může mít jinou šířku pro každý breakpoint
- Hlavní výhoda RWD = jiný layout pro každý breakpoint

# Jinak široký sloupec dle zařízení



HTML

```
<div class="col-6 col-sm-4 col-md-3">
```

```
<!-- 1/2 mobil, 1/3 tablet, 1/4 pc -->
```

```
</div>
```



# Šířka sloupce se dědí směrem nahoru



- Nemusíme definovat šířku pro každý breakpoint
- Pokud není explicitně řečeno jinak, je šířka stejná jako u předchozího breakpointu

## Jinak široký sloupec dle zařízení



HTML

```
<div class="col-6 col-lg-4">
```

```
<!-- "chybí" sm, md, xl -->
```

```
</div>
```

PROJEKT