

# BGWE1

Mgr. Vlastislav Kučera, Ph.D.

grid system

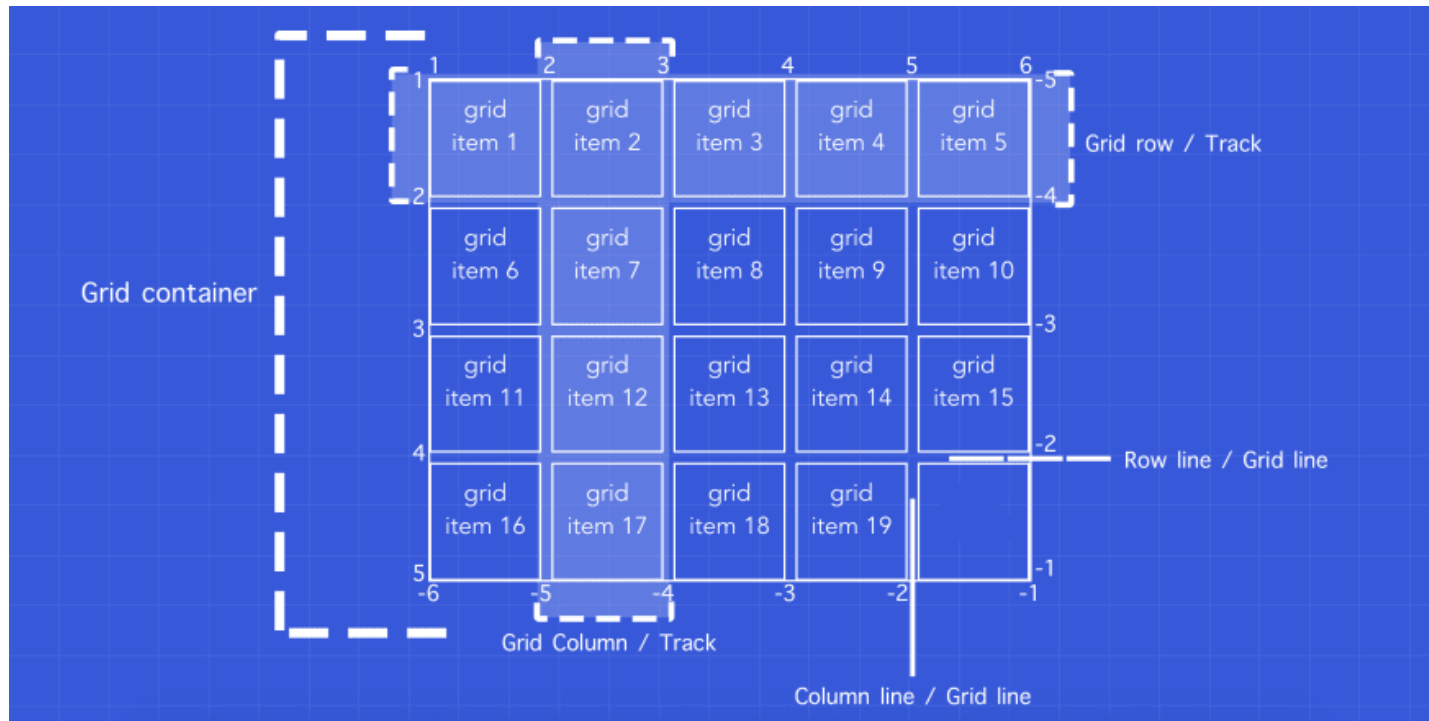
# grid system

- ▶ nový způsob rozvržení (layoutu) stránky
- ▶ dá se říct, že do tvorby webu byl implementován způsob pro návrh a rozvržení stránky pro tisk a digitální média, např. Adobe InDesign
- ▶ pomocí níže uvedených vlastností se nadefinuje mřížka stránky, do které se následně vloží jednotlivé části stránky (záhlaví, obsah, zápatí, ...)
- ▶ podpora
  - ▶ všechny aktuální verze prohlížečů => můžeme grid používat

grid - obrázek

zdroj:

[https://res.cloudinary.com/practicaldev/image/fetch/s--\\_YJvL5YJ--/c\\_limit%2Cf\\_auto%2Cfl\\_progressive%2Cq\\_auto%2Cw\\_880/https://i.ibb.co/dkYpKRT/grid.jp](https://res.cloudinary.com/practicaldev/image/fetch/s--_YJvL5YJ--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https://i.ibb.co/dkYpKRT/grid.jp)  
g



# grid - příklad

```
<!DOCTYPE html>  
  <html lang='cs'>  
    <head>  
    </head>  
    <body>  
      <div id="stranka">  
        <header class="logo">Záhlaví</header>  
        <div id="obsah">Obsah</div>  
        <nav class="nav">Navigace</nav>  
        <footer class="pata">Zápatí</footer>  
      </div>  
    </body>  
  </html>
```

# grid system

- ▶ abychom mohli grid využít, musíme u prvku, který budeme chtít pomocí gridu rozdělit, zapnout příslušný způsob zobrazení
- ▶ zapnutí gridu se provede pomocí příkazu `display: grid`
- ▶ příklad využívá zápis z předchozího snímku:
  - ▶ řekněme, že prvek `<div id="stranka">` má být zobrazen v grid režimu, jinak řečeno, prvek má svůj obsah zobrazit pomocí mřížky
  - ▶ ve stylovém předpisu nastavíme následující:
  - ▶ `div#stranka {display: grid}`

# grid system

- ▶ kromě samotného zobrazení mřížky (display: grid), musíme ještě definovat sloupce a řádky pomocí následujících vlastností
- ▶ u uvedených vlastností se definuje počet sloupců/ řádků tím, že uvedeme šířky jednotlivých sloupců/ výšky řádků
- ▶ grid-template-columns
  - ▶ definice sloupců
- ▶ grid-template-rows
  - ▶ definice řádků

# šířka sloupce (řádku)

- ▶ zadává se jednotka
- ▶ jednotka
  - ▶ podobně jako při definici velikosti prvku, velikosti písma, ...
  - ▶ px, em, rem, %, auto
  - ▶ fr - jednotka volného místa v mřížce
  - ▶ např. grid-template-columns: 3fr 2fr
    - ▶ celková šířka se rozdělí na pět stejných částí, z toho 3 případnou prvnímu sloupci, 2 druhému
- ▶ můžeme použít i následující funkce
  - ▶ minmax()
  - ▶ repeat()

# minmax()

- ▶ slouží k zadání minimální a maximální velikosti šířky/ výšky
- ▶ zadávají se dvě hodnoty: `minmax(50px, 50%)`
  - ▶ 1. hodnota určuje minimální šířku/výšku, druhé číslo maximální
  - ▶ uvedený příklad říká, že sloupec bude široký nejméně 50px a nejvýše 50% celkové šířky mřížky (obdobně pro řádky)
- ▶ `grid-template-columns: minmax(auto, 33%) 2fr`
  - ▶ 1. sloupec bude mít automatickou šířku, ale maximální 33% šířky mřížky



# repeat()

- ▶ vhodné při definování sloupců/řádků se stejnou velikostí
- ▶ dvě hodnoty: `repeat(3, 1fr)`
  - ▶ první hodnota - počet opakování
  - ▶ druhá hodnota - velikost
  - ▶ `grid-template-columns: repeat(3, 1fr) = grid-template-columns: 1fr 1fr 1fr`

# grid-column-gap

## grid-row-gap

- ▶ slouží k definování rozestupů mezi jednotlivými políčky mřížky (grid item z obrázku na snímku 3)
- ▶ grid-column-gap
  - ▶ rozestup ve sloupcích
- ▶ grid-row-gap
  - ▶ rozestup v řádcích
- ▶ zadává se velikost (px, em, %, ...)

# grid-gap

- ▶ sdružená vlastnost pro definování rozestupu
- ▶ 1 nebo 2 hodnoty
  - ▶ 1 hodnota - stejná pro rozestup sloupců i řádků
  - ▶ 2 hodnoty - první - rozestup mezi řádky, druhá - rozestup mezi sloupci

# zarovnávání sloupců

- ▶ zadává se u prvku, který má definován display: grid
- ▶ justify-content
- ▶ hodnoty:
  - ▶ normal
  - ▶ start - zarovná sloupce od začátku řádku
  - ▶ end - zarovná sloupce na konce řádku
  - ▶ center - sloupce vycentruje
  - ▶ stretch
  - ▶ space-around - kolem sloupců stejné mezery
  - ▶ space-between - mezi sloupci stejné mezery
  - ▶ space-evenly
  - ▶ baseline
  - ▶ first baseline
  - ▶ last baseline

# zarovnávání řádků

- ▶ align-content
- ▶ hodnoty - stejné jako u justify-content

# grid - definování oblasti

- ▶ `grid-column-start`
  - ▶ začátek oblasti ve vodorovném směru
- ▶ `grid-column-end`
  - ▶ konec oblasti ve vodorovném směru
- ▶ `grid-row-start`
  - ▶ začátek oblasti ve svislém směru
- ▶ `grid-row-end`
  - ▶ konec oblasti ve svislém směru

# grid - definování oblasti

- ▶ hodnoty
  - ▶ čísla
- ▶ čísla značí jednotlivé mřížky („čáry“ mezi jednotlivými políčky (grid item z obrázku na snímku 3), nikoliv oblasti)
- ▶ `grid-column-start: 2; grid-column-end: 5`
  - ▶ oblast začíná na druhé „čáře“ zleva a končí na páté
- ▶ nebo můžeme zadat
  - ▶ `grid-column-start: 2; grid-column-end: span 3`
    - ▶ oblast začíná na druhé „čáře“ a končí na třetí od zadané pomocí `grid-column-start`, tj. na třetí od druhé
- ▶ totéž platí i pro `grid-row-start(end)`

# grid - definování oblasti

- ▶ oblasti můžeme definovat i zprava
  - ▶ `grid-column-start: 5; grid-column-end: 2`



# grid - definování oblasti

- ▶ sdružené vlastnosti:
  - ▶ grid-column, grid-row
    - ▶ hodnoty odděleny „/“
    - ▶ hodnota před „/“ odpovídá grid-column(row)-start
    - ▶ hodnota za „/“ odpovídá grid-column(row)-end
    - ▶ když je jedna hodnota, je použita pro grid-column(row)-start
  - ▶ grid-area - sdružená vlastnost pro grid-column a grid-row
    - ▶ 4 hodnoty odděleny „/“
    - ▶ 1. hodnota = grid-row-start
    - ▶ 2. hodnota = grid-column-start
    - ▶ 3. hodnota = grid-row-end
    - ▶ 4. hodnota = grid-column-end

# grid - příklad, html kód

```
<!DOCTYPE html>
  <html lang='cs'>
    <head>
    </head>
    <body>
      <div id="stranka">
        <header class="logo">Záhlaví</header>
        <div id="obsah">Obsah</div>
        <nav class="nav">Navigace</nav>
        <footer class="pata">Zápatí</footer>
      </div>
    </body>
  </html>
```

# grid-příklad

- ▶ mějme následující rozložení stránky:

	Header
Menu	Hlavní obsah
	Patka

# definice pomocí již známých vlastností: grid-column-start (end), grid-row-start(end)

```
#stranka{display: grid;
```

```
  grid-template-columns: 1fr 6fr;
```

```
  grid-template-rows: 120px auto 30px;
```

```
  grid-gap: 10px;}
```

```
header{grid-column-start: 1; grid-column-end: 3; grid-row-start: 1; grid-row-end: 2;}
```

```
nav{grid-column-start: 1; grid-column-end: 2; grid-row-start: 2; grid-row-end: 3;}
```

```
#obsah{grid-column-start: 2; grid-column-end: 3; grid-row-start: 2; grid-row-end: 3;}
```

```
footer{grid-column-start: 1; grid-column-end: 3; grid-row-start: 3; grid-row-end: 4;}
```

# definice pomocí již známých vlastností: grid-column, grid-row, ...

```
#stranka{display: grid;  
    grid-template-columns: 1fr 6fr;  
    grid-template-rows: 120px auto 30px;  
    grid-gap: 10px;}  
header{grid-column: 1/3; grid-row: 1/2;}  
nav{grid-column: 1/2; grid-row: 2/3;}  
#obsah{grid-column: 2/3; grid-row: 2/3;}  
footer{grid-column: 1/3; grid-row: 3/4;}
```

# definice pomocí již známých vlastností: grid-area

```
#stranka{display: grid;  
    grid-template-columns: 1fr 6fr;  
    grid-template-rows: 120px auto 30px;  
    grid-gap: 10px;}  
header{grid-area: 1/1/2/3;}  
nav{grid-area: 2/1/3/2;}  
#obsah{grid-area: 2/2/3/3;}  
footer{grid-area: 3/1/4/3;}
```

# grid-template-areas

- ▶ slouží k definování mřížky pomocí pojmenování oblastí

# definice pomocí již známých vlastností: grid-column, grid-row, ...

```
#stranka{display: grid;
```

```
  grid-template-columns: 1fr 6fr;
```

```
  grid-template-rows: 120px auto 30px;
```

```
  grid-gap: 10px;}
```

```
header{grid-column-start: 1; grid-column-end: 3; grid-row-start: 1; grid-row-end: 2;}
```

```
nav{grid-column-start: 1; grid-column-end: 2; grid-row-start: 2; grid-row-end: 3;}
```

```
#obsah{grid-column-start: 2; grid-column-end: 3; grid-row-start: 2; grid-row-end: 3;}
```

```
footer{grid-column-start: 1; grid-column-end: 3; grid-row-start: 3; grid-row-end: 4;}
```



# definice pomocí grid-template-areas

```
#stranka{display: grid;
  grid-template-columns: 1fr 6fr;
  grid-template-rows: 120px auto 30px;
  grid-gap: 10px;
  grid-template-areas:
    "logo logo"
    "nav obsah"
    "pata pata";}
header{grid-area: logo;}
nav{grid-area: nav;}
#obsah{grid-area: obsah;}
footer{grid-area: pata;}
```

# grid

- ▶ Pro definici mřížky používejte ten způsob, který je pro vás nejlépe pochopitelný
- ▶ je jen na vás, jestli budete používat `grid-column-start`, `grid-column-end`, ..., nebo `grid-column (row)`, nebo `grid-area` nebo dokonce `grid-template-areas`

# grid - hra

- ▶ hra pro výuku grid systemu: <http://cssgridgarden.com/#cs>

# grid - zdroje

- ▶ generátory gridu:
  - ▶ <https://cssgrid-generator.netlify.com/>
  - ▶ <https://grid.layoutit.com/>
- ▶ <https://learncssgrid.com/>
- ▶ <https://css-tricks.com/snippets/css/complete-guide-grid/>
- ▶ <https://www.joshwcomeau.com/css/interactive-guide-to-grid/>
- ▶ <https://www.v-kucera.cz/?tag=grid>