

HTML CSS

Webová grafika

Marcel Makovník

KAG, FMFI UK

Základná štruktúra webstránky

Obsahová časť

- Je písaná v jazyku **HTML** (hypertext markup language)
- Súbory majú koncovku *(.html)*
- Pomocou HTML zapisujeme, čo stránka obsahuje a jej štruktúru

Formálna časť

- Je písaná v jazyku **CSS** (cascading style sheets)
- Súbory majú koncovku *(.css)*
- Pomocou CSS zapisujeme vzhľad a umiestnenie jednotlivých objektov stránky

Základná štruktúra html dokumentu

- Jednotlivé objekty sú ohraničené značkami (tagmi) vo formáte

```
<div class="clanok">Obsah článku</div>
```

Názov tagu
(začiatok)

Názov vlastnosti
tagu

Hodnota
vlastnosti

Obsah tagu

Koniec tagu

- Každému typu objektu prislúcha osobitný tag
- Väčšina objektov má párové tagy (t. j. nachádzajú sa aj na začiatku aj na konci)
 - Napr. `<p>Odstavec textu</p>` - nám vypíše text „Odstavec textu“
- Tagy slúžia aj na vytvorenie samotnej štruktúry dokumentu

Základná štruktúra html dokumentu

Tento riadok oznámi prehliadaču, že zvyšok kódu je písaný v jazyku HTML

```
<!DOCTYPE html>
```

Vo vnútri tohto tagu sa nachádza obsah stránky

```
<html>
```

Hlavička dokumentu – tu zadávame všetky charakteristiky dokumentu, názov stránky a odkazy na externé súbory.

```
<head>
```

```
<title>Názov stránky</title>
```

```
<meta charset="utf-8">
```

```
<link href="style.css" rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
<h1>Nadpis</h1>
```

```
<p>Text k nadpisu</p>
```

```
</body>
```

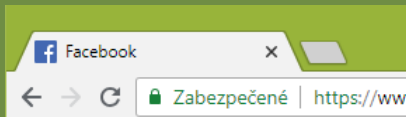
```
</html>
```

Telo dokumentu – všetky tagy vo vnútri tela sa vyobrazia v okne webového prehliadača

Hlavička dokumentu

- V každej hlavičke by sa mali nachádzať aspoň tieto tri tagy:

Názov stránky – zobrazuje sa na hornej lište prehliadača, napr.



Súprava použitých znakov v dokumente. Hodnota „utf-8“ nám zabezpečí, že sa budú zobrazovať správne písmena s dĺžňami, mäkčeňmi a iné špeciálne symboly

Načítavanie externých súborov, v našom prípade je to (.css) súbor, ktorým formátujeme dokument

```
<head>
```

```
  <title>Názov stránky</title>
```

```
  <meta charset="utf-8">
```

```
  <link href="style.css" rel="stylesheet">
```

```
</head>
```

TAGY V TELE DOKUMENTU

Odstavce textu a nadpisy

- Text je vhodné formátovať do odstavcov nasledovne

```
<p>  
    Prvý riadok odstavca <br>  
    Druhý riadok odstavca  
</p>
```

- Pre nadpisy máme k dispozícii šesť úrovní

```
<h1>Najväčší nadpis</h1>  
.  
.  
.  
<h6>Najmenší nadpis</h6>
```

Tag `
` nás prepne do nového riadku v rámci odstavca.

Všimnime si, že tento tag je nepárový, t.j. nepoužíva sa koniec tagu `</br>`

Zvýraznenie textu

- Ak chceme v rámci odstavca nejakú časť textu zvýrazniť tučným písmom alebo kurzívou, použijeme nasledovné tagy

Text vo vnútri tagu ``
sa zobrazí nasledovne:

...a *kurzivu*.

`<p>V našom texte chceme použiť tučné písmo a kurzivu.</p>`

Text vo vnútri tagu
`` sa
zobrazí nasledovne:

...použiť **tučné písmo** a...

Hypertextové odkazy

- Na našej stránke chceme vedieť preklikávať medzi jednotlivými podstránkami, resp. sa presmerovať na inú webstránku, alebo súbor
- Na tento účel nám poslúžia hypertextové odkazy, ktorým prislúcha tag `<a>`

Ako hodnotu atribútu href uvádzame cestu, kam sa chceme presmerovať.

Môžeme sa odkazovať na (.html) súbor, čiže podstránku, ale taktiež aj URL adresu, alebo akýkoľvek iný typ súboru (takto sa vytvorí odkaz na stiahnutie)

```
<a href="podstranka.html">Klikni sem.</a>
```

Text vo vnútri tagu sa zobrazí ako klikateľný odkaz. Zvyčajne nadobudne nasledovné zvýraznenie:

[Klikni sem.](#)

Obrázky

- Pre obrázky sa používa nepárový tag s nasledovnými dvoma povinnými atribútmi

V atribúte alt uvidíme stručný popis obrázku.

Hodnota tohto atribútu sa zobrazí na stránke v prípade, že sa obrázok nepodarí zo súboru načítať.

```

```

V atribúte src uvidíme cestu k obrázku.

Obrázky môžu byť vo formátoch napr. (.jpg, .png, .svg, .gif).

CSS

Webová grafika

Marcel Makovník

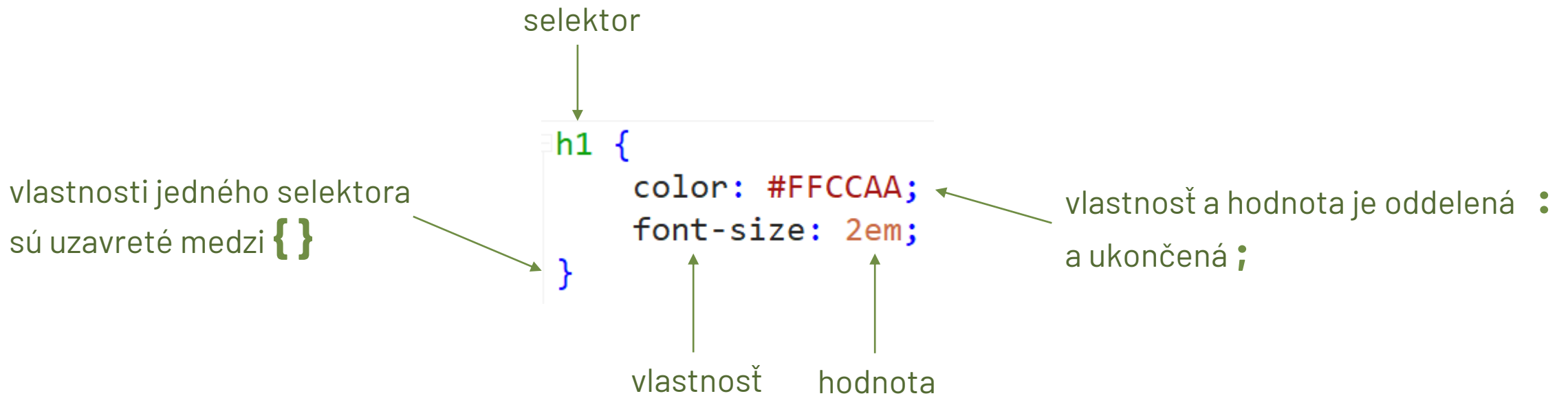
KAG, FMFI UK

CSS

- Opisuje vzhľad jednotlivých HTML elementov a ako majú byť zobrazené
- Je to osobitný (.css) súbor
- Môžeme teda pomocou neho formátovať viacero stránok naraz
- Prepojenie (.html) a (.css) súboru

```
<link href="styly/styl.css" type="text/css" rel="stylesheet">
```

CSS - syntax



CSS – typy selektorov

element selektor

- formátuje všetky elementy označené príslušným tagom

id selektor

- vyberá element na základe „id“ atribútu
- tento element by mal jedinečný pre celú stránku

class selektor

- vyberá element na základe „class“ atribútu
- formátuje všetky elementy s rovnakou hodnotou „class“

HTML:

```
<h1>Nadpis</h1>
```

CSS:

```
h1 {  
  color: #FFCCAA;  
  font-size: 2em;  
}
```

HTML:

```

```

CSS:

```
#logo {  
  height: 80px;  
}
```

HTML:

```

```

CSS:

```
.ikona {  
  margin: 10px 10px 10px 10px;  
}
```

Text

- farba
 - **color:** HEX (#FF0000), RGB (rgb(255, 255, 255)), RGBA(rgba(255, 255, 255, 1.0))
- zarovnanie textu
 - **text-align:** left, right, center, justify (natiehnutie riadku na celú šírku)
- rozostup medzi písmenami
 - **letter-spacing:** hodnota dĺžky (px, em, pt, % ...)
- výška riadka
 - **line-height:** hodnota dĺžky (px, em, pt, % ...)
- zvýraznenie textu
 - **text-decoration:** none (najmä pri linkoch), underline, line-through
- transformácia textu
 - **text-transform:** uppercase (všetko veľkými písmenami), lowercase, capitalize (prvé veľkým)

Písmo

- rodina písma
 - **font-family**: 'názov písma', serif, sans-serif, monospace
- veľkosť písma
 - **font-size**: hodnota dĺžky, odporúča sa jednotka em (násobok predvolenej veľkosti písma v prehliadači, väčšinou 16px), prípadne %, pt (toto je hlavne pre tlač)
- hrúbka písma
 - **font-weight**: 100, ..., 900, bold (700), normal (400)

1 Family Selected

Your Selection [Clear All](#)



Montserrat Alternates

EMBED

CUSTOMIZE

Load Time: Fast

Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD @IMPORT

```
<link href="https://fonts.googleapis.com/css?family=Montserrat+Alternates:400,700" rel="stylesheet">
```

Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Montserrat Alternates', sans-serif;
```

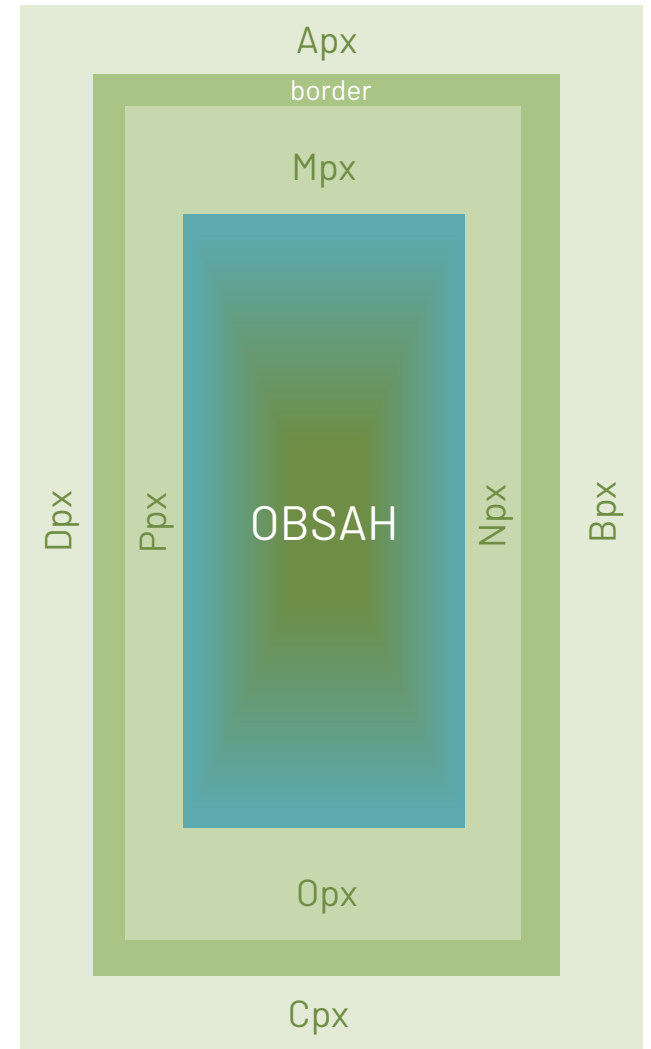
For examples of how fonts can be added to webpages, see the [getting started guide](#).

Vloženie webfontu na stránku

- podľa pokynov služby, ktorú použijeme
- väčšinou stačí skopírovať jeden riadok do HTML súboru
- v CSS súbore sa odvolávame už len na príslušný názov písma

Rozmery a okraje

- šírka / výška
 - **width / height**: hodnota dĺžky (px, em, pt, % ...)
- vonkajší okraj
 - **margin**: Apx Bpx Cpx Dpx; (prípadne iná hodnota dĺžky)
 - **margin-top**: Apx; **margin-right**: Bpx; **margin-bottom**: Cpx; **margin-left**: Dpx;
- vnútorný okraj
 - **padding**: Mpx Npx Opx Ppx; (prípadne iná hodnota dĺžky)
 - **padding-top**: Mpx; **padding-right**: Npx; **padding-bottom**: Opx; **padding-left**: Ppx;

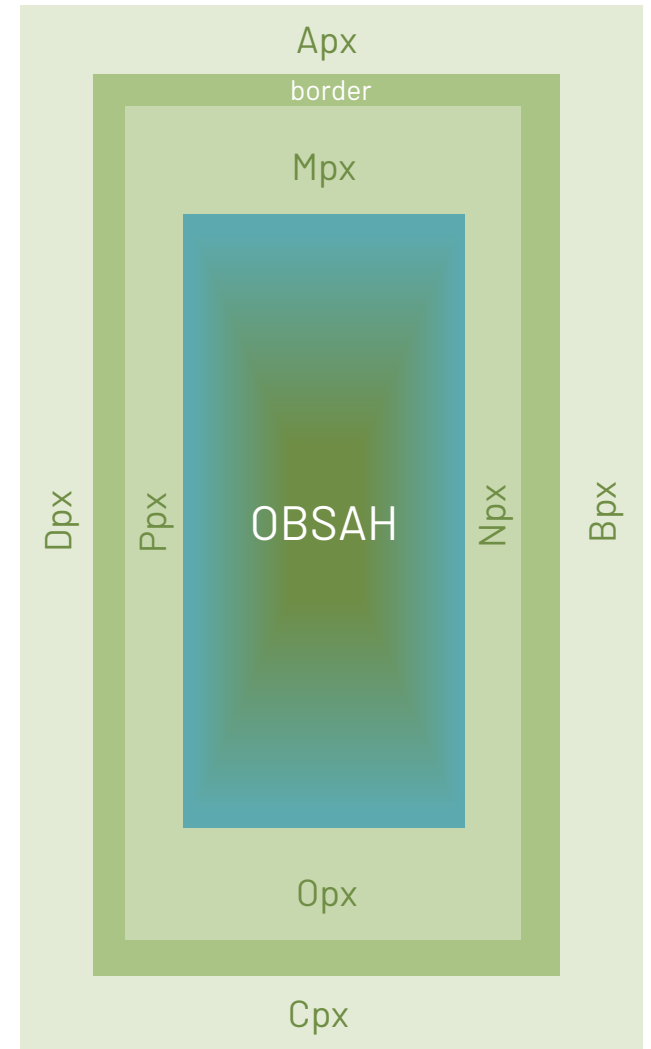


Rozmery a okraje

- ohraničenie **border**:
 - **border-style**: solid, dotted, dashed, none... (vždy treba špecifikovať ak chceme zobrazit ohraničenie)
 - **border-width**: hodnota dĺžky (px, em, pt, % ...)
 - **border-color**: HEX (#FF0000), RGB (rgb(255, 255, 255)), RGBA(rgba(255, 255, 255, 1.0))
- každú vlastnosť môžeme nastaviť súhrnne pre všetky štyri strany (jedna hodnota), alebo individuálne pre jednotlivé strany (štyri hodnoty)

```
border-width: 5px;
```

```
border-width: 2px 0 2px 0;
```



ROZLOŽENIE

Webovská grafika

Marcel Makovník

KAG, FMFI UK

Rozloženie (layout)

- Webstránku vieme rozložiť na logické časti
 - t.j. – vieme určiť, kde sa nachádza hlavička, kde sa nachádza obsah, navigácia
- Pred HTML5 – `<div class="nazov casti">Obsah</div>`
- HTML5 – osobitné tagy pre jednotlivé časti stránky



HTML5 – prvky rozloženia

- `<header>`
 - hlavička dokumentu alebo sekcie
- `<nav>`
 - navigácia, zoznam odkazov na iné stránky
- `<section>`
 - definuje sekciu (časť) dokumentu
- `<article>`
 - samostatný, nezávislý článok v rámci dokumentu
- `<aside>`
 - obsah „pomimo“, bočný panel
- `<footer>`
 - päta dokumentu

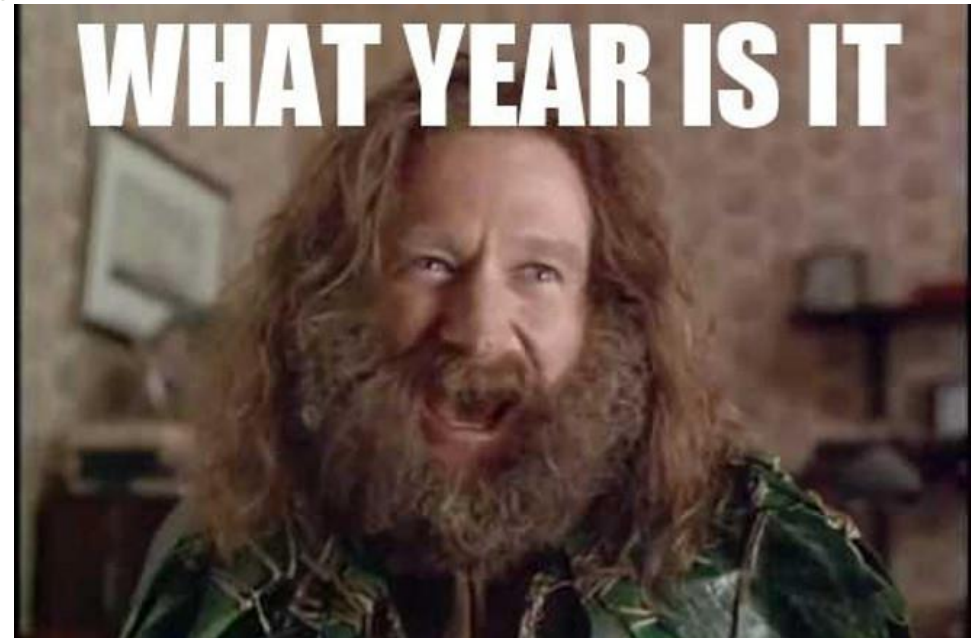
Techniky tvorby layoutu

- jednotlivé časti dokumentu treba „naaranžovať“
- základné techniky sú
 - HTML tabuľky
 - CSS float
 - frameworky
 - CSS flexbox/grid

HTML tabuľky

- na jednotlivé časti stránky sa používajú tagy <table>, <tr>, <td> a tie formátujeme
- tieto tagy slúžia na zobrazovanie tabulárnych dát
- takto sa robili layouts pred rokom....

TABUĽKY SA
NEPOUŽÍVAJÚ NA
TVORBU LAYOUTU!!!



CSS float – obtekanie

- bežne používaný postup
- formátujeme pomocou CSS vlastností float a clear
- vysvetlenie, návod a ukážka jednoduchého layoutu pomocou float a clear
- https://www.w3schools.com/css/css_float.asp
- nevýhodou je, že pomocou float a clear nedokážeme meniť poradie jednotlivých prvkov v dokumente (flow)

Frameworky

- majú nám uľahčiť tvorbu webu
- filozofická otázka – ako veľmi?

- Bootstrap, W3.CSS, Foundation atď.

- výhody:
 - mobile-first layout, „dobrá“ stránka za krátky čas, cross-browser, ...
- nevýhody:
 - neprehľadný kód (?), neunikátnosť, učiť sa ďalšiu terminológiu, ...

Frameworky

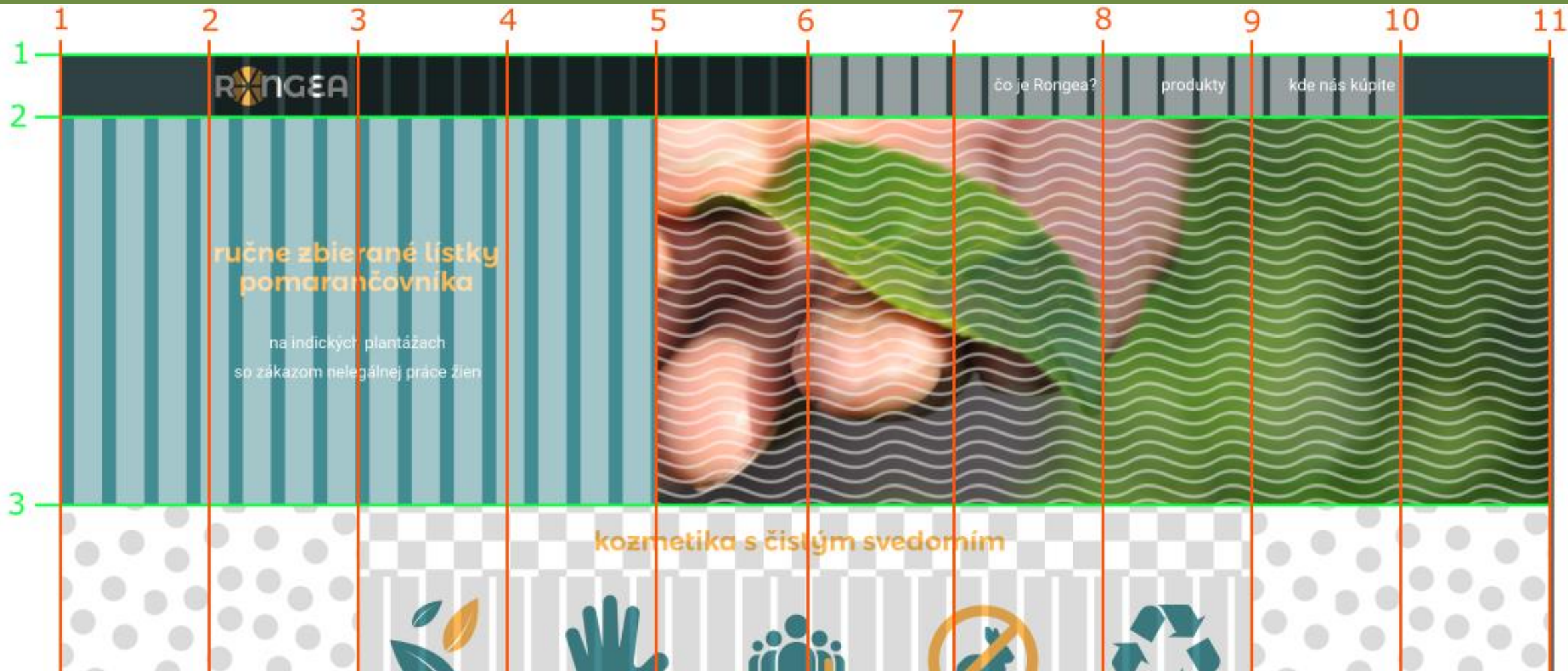
- v konečnom dôsledku:
 - na pochopenie fungovania frameworku treba aspoň základnú znalosť CSS, resp. JavaScript
 - vlastné modifikácie sa (väčšinou) implementujú ťažšie

NA CVIČENIACH A V PROJEKTOCH PREDMETU WEBOVSKÁ GRAFIKA SA
PRETO FRAMEWORKY POUŽÍVAŤ NEBUDÚ.

- frameworky sú super a v praxi sa im (zrejme) nevyhnete, ale...

CSS flexbox/grid

- floaty pôvodne neboli určené na tvorbu layoutu
- flexbox (resp. grid) áno a fungujú intuitívnejšie
- flexbox
 - 1D layout, dlhšia podpora
- grid
 - 2D layout, relatívne krátka podpora (nové)
- flexbox a grid (pravdepodobne) budú fungovať jeden vedľa druhého



CSS GRID

mriežka, ktorá nám pomáha pri rozvrhnutí stránky

CSS grid – kontajner

- miesto, kde nastavujeme šírku stĺpcov, výšku riadkov a pozíciu jednotlivých buniek, t.j. mriežku

```
body
{
  display: grid;

  /* šírka stĺpcov */
  grid-template-columns: 20% 20% 20% 20% 20%;

  /* výška stĺpcov */
  grid-template-rows: 100px 80px 150px;

  /* umiestnenie mriežky v rámci kontajnera */
  justify-content: center | start | end | space-evenly | -around | -between;

  /* umiestnenie buniek v rámci mriežky */
  align-content: center | start | end | space-evenly | -around | -between;
}
```

CSS grid – a) každej bunke určíme pozíciu

```
header{
```

```
  grid-column-start: 2;  
  grid-column-end: 6;  
  grid-row-start: 1;  
  grid-row-end: 2;
```

```
  grid-row: 1 / 2;  
  grid-column: 2 / 6;
```

```
  grid-row: 1 / span 1;  
  grid-column: 2 / span 4;
```

```
  grid-area: 1 / 2 / 2 / 6;
```

```
}
```

- každá bunka má tvar obdĺžnika
- zadávame číslo priamky ľavého horného rohu a pravého dolného rohu
- 4 rôzne spôsoby – každý robí to isté
- poradie v HTML a CSS môže byť rôzne
- veľmi výhodné pre media queries

CSS grid – b) vytvoríme template v kontajneri

- namiesto vypisovania začiatku a konca pre jednotlivé bunky môžeme zapísať rozloženie buniek do template

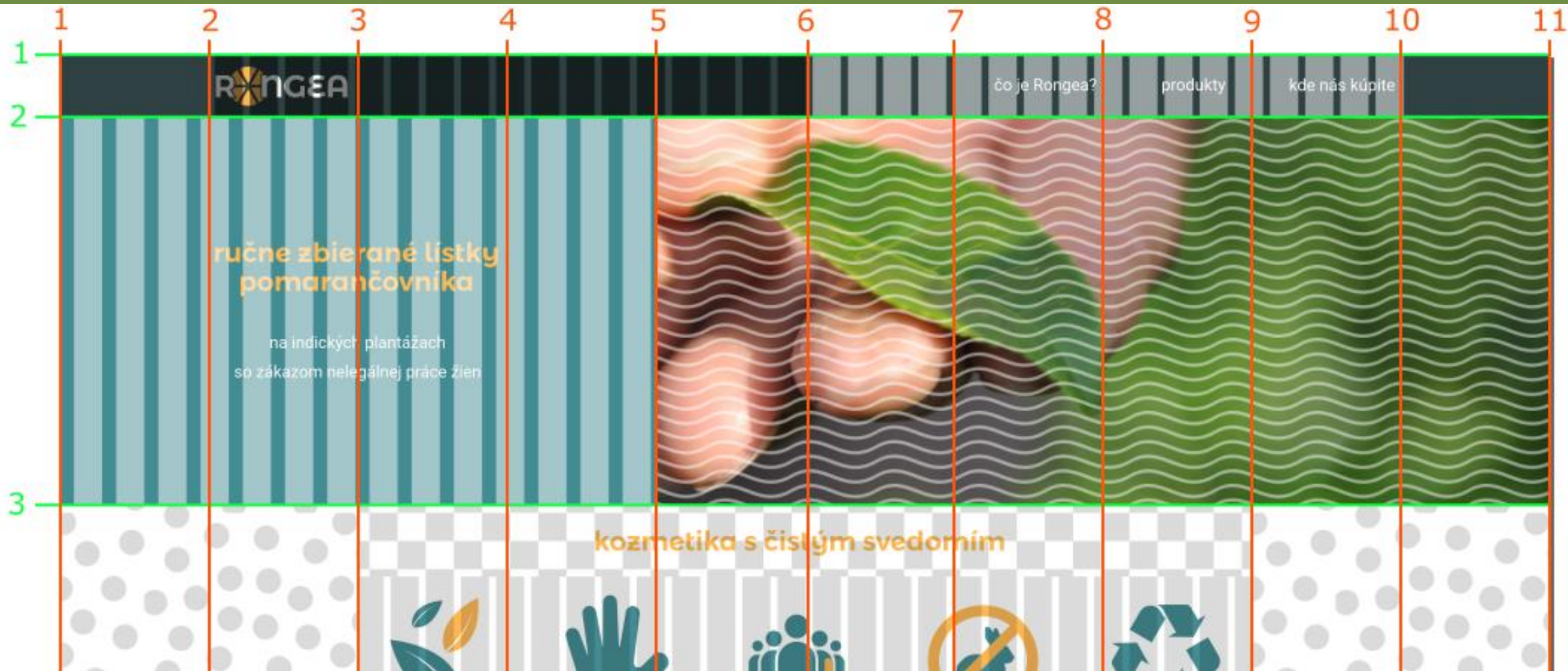
```
.red{  
  background-color: rgb(200, 50, 50);  
  color: rgb(255, 255, 255);  
  font-size: 36pt;  
  
  grid-area: red-container;  
}
```

```
.blue{  
  background-color: rgb(50, 50, 150);  
  color: rgb(255, 255, 255);  
  font-size: 36pt;  
  
  grid-area: blue-container;  
}
```

```
.green{  
  background-color: rgb(50, 150, 50);  
  color: rgb(255, 255, 255);  
  font-size: 36pt;  
  
  grid-area: green-container;  
}
```

```
.grey{  
  background-color: rgb(50, 50, 50);  
  color: rgb(255, 255, 255);  
  font-size: 36pt;  
  
  grid-area: grey-container;  
}
```

```
body{  
  
  grid-template-areas:  
    'red-container      red-container      red-container      red-container'  
    'blue-container     grey-container     grey-container     .'  
    'green-container    green-container    green-container    green-container';  
}
```

CSS FLEXBOX

keď floaty nefungujú tak, ako by sme chceli

CSS flexbox – kontajner

```
display: flex;
```

```
/* smer a zalamovanie ako dve vlastnosti */
```

```
flex-direction: column | column-reverse | row | row-reverse;
```

```
flex-wrap: nowrap | wrap | wrap-reverse;
```

```
/* tok - smer a zalamovanie v jednom */
```

```
flex-flow: row wrap;
```

```
/* zarovnanie jednotlivých položiek (vodorovne) */
```

```
justify-content: center | start | end | space-around | space-between;
```

```
/* zarovnanie položiek (zvisle) */
```

```
align-items: flex-start | flex-end | center | stretch | baseline;
```

- vieme dosiahnuť dokonalé centrovanie (?)

CSS flexbox – položky

- podrobne:
 - https://www.w3schools.com/css/css3_flexbox.asp

- jedna dôležitá vlastnosť – vieme meniť poradie prvkov

```
.prod-1{  
  order: 1;  
}
```



```
.prod-2{  
  order: 2;  
}
```



```
.prod-3{  
  order: 3;  
}
```



```
.prod-1{  
  order: 2;  
}
```



```
.prod-2{  
  order: 3;  
}
```



```
.prod-3{  
  order: 1;  
}
```



Zdroje a užitečné linky

- https://www.w3schools.com/html/html_layout.asp
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>