

## CSS GRID – koniec bolehlavu pri tvorbe layoutu

### Úvod:

V mojom článku som sa rozhodol stručne popísať skúsenosti s tzv. css Grid-om, aby som ti pomohol “vložiť” pri tvorbe tvojej webstránky to čo chceš, práve tam kam chceš, ušetril ti hodiny nervov so systémom pokus omyl a predĺžil tak životnosť klávesovej skratky CTRL+R. Takže ti prajem príjemné čítanie a snáď ťa na konci môjho článku presvedčím, že to fakt

“nebolí”

### Čo je CSS GRID a na čo nám slúži?

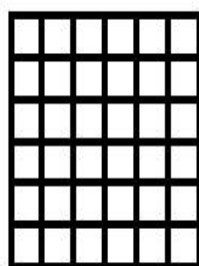
Kdekoľvek na internete nájdeme mnoho “definícií” gridu ako napr: Rozvrhnutie mriežky CSS GRID je najvýkonnejší systém rozloženia dostupný v CSS. Je to dvojrozmerný systém, čo znamená, že zvláda stĺpce aj riadky. S rozložením mriežky pracujeme tak, že pravidlá CSS aplikujeme na nadradený prvok (ktorý sa stáva mriežkou). (zdroj: <https://css-tricks.com/snippets/css/complete-guide-grid>)

Samozrejme definície nik z nás neobľubuje a preto som sa rozhodol grid objasniť aj tebe bez toho aby si potreboval znalosti jadrového fyzika, podľa teda nato:

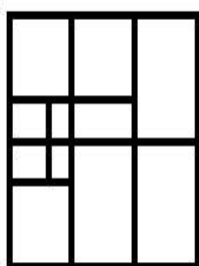
- ak by som chcel grid jednoducho charakterizovať, mohol by som ho prirovnať k situ
- v závislosti čo idem preosievať si musím vybrať sito so zodpovedajúcou veľkosťou okien – pomenujeme diery v site pre náš prípad **okná** a sito pomenujeme **mriežkou**, keďže tieto pojmy budem v mojom článku používať častejšie
- tak teda v závislosti čo idem na mojej webstránke zobrazovať si vyberiem (zhotovím) mriežku do ktorej potom budem vkladať jednotlivé elementy (obrázky, texty, navigácie, odkazy,proste úplne čokoľvek....!dokonca môžem vnárať mriežku do mriežky!) – možnosti sú nekonečné, brzdiť ťa nebude nič len tvoja fantázia.

V skratke: css grid je teda **dvojrozmerná mriežka** v ktorej pracujeme s **riadkami (row)** a **stĺpcami (column)** (dalo by sa povedať ako tabuľka s jednotlivými poliami)

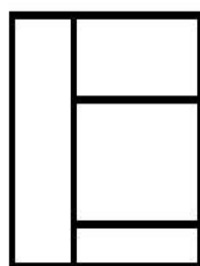
Tabuľky nie sú nikdy rovnaké s úplne rovnakými poliami, tak si aj jednotlivé okná gridu dokážem prispôbiť mojim potrebám :



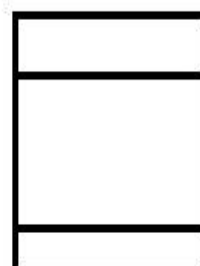
Ahoj ja som GRID



Ahoj ja som GRID



Ahoj ja som GRID



Ahoj ja som GRID

## Teória je síce pekná a čo prax?

K vytvoreniu GRID layoutu (rozloženia mriežok) môžeme pristupovať viacerými spôsobmi – ja ti ukážem dva ktoré sú podľa mňa najjednoduchšie.

### PRVÝ spôsob:

vytvorím si html:

```
<div class="mriezka">
  <div class="okna okno1">
  </div>
  <div class="okna okno2">
  </div>
  <div class="okna okno3">
  </div>
  <div class="okna okno4">
  </div>
  <div class="okna okno5">
  </div>
  <div class="okna okno6">
  </div>
</div>
```

- vytvorím element mriežka – ktorý je nadradeným elementom okna, čiže podľa definície **mriežka** bude mriežkou (mojim sitom) a **okna** budú jednotlivé dierky v mojom site
- jednotlivým elementom **okna** som pridal ďalšiu class **okno1, okno2 ....** aby som si ich neskôr mohol upraviť ľubovoľne každé okno zvlášť

vložím prvé css:

```
.mriezka {
  display: grid;
  grid-template-columns: repeat(8, 1fr);
  grid-template-rows: repeat(8, 5vw);
  grid-gap: 1vw;
}

.okna {
  background-color: black;
}
```

- môjmu gridu pridám css:
- **display: grid;** – tento príkaz nám hovorí že element mriežka má byť prevedený do gridu – !tento príkaz platí pre všetky druhy zobrazenia gridu – **proste sa ho nauč!**
- **grid-template-columns: repeat(8, 1fr);** – zobraz mi element mriežka tak aby mal 8 stĺpcov, každý veľkosti **1fr = 1 fraction** (nezáleží na veľkosti, proste aby boli všetky úplne rovnaké, tu by som mohol použiť aj konkrétne hodnoty ako 100px, 10% a pod ale **fr** mi zabezpečí, že je jedno na akom zariadení zobrazujem daný grid, či na pc či na mobile vždy bude element mriežka rozdelený na 8 rovnakých stĺpcov
- **grid-template-rows: repeat(8, 5vw);** – zobraz mi element mriežka taky aby mal 8 riadkov, kde každý riadok bude mať veľkosť 5 vw (5 view width) – znovu by som mohol použiť hodnoty ako px ... ale keďže my sme moderní a chceme aby naša mriežka bola responzívna ( v závislosti na akom zariadení moju mriežku pozerám) a tým aby sme boli IN budem používať hodnoty ako fr a vw
- **grid-gap: 1vw;** vlož mi medzeru veľkosti 1vw (view width) medzi jednotlivé okná mojej mriežky
- **background-color: black;** – vyfarbi jednotlivé okná na čierne (aby som videl s čím pracujem)



Mám 6 okien ktoré zatiaľ nemajú zadefinovanú polohu

úprava jednotlivých okien:

```
.okno1 {
  grid-column-start: 1;
  grid-column-end: 3;
  grid-row-start: 1;
  grid-row-end: 3;
}

.okno2 {
  grid-column-start: 3;
  grid-column-end: 5;
  grid-row-start: 1;
  grid-row-end: 3;
}

.okno3 {
  grid-column-start: 5;
  grid-column-end: 9;
  grid-row-start: 1;
  grid-row-end: 6;
}

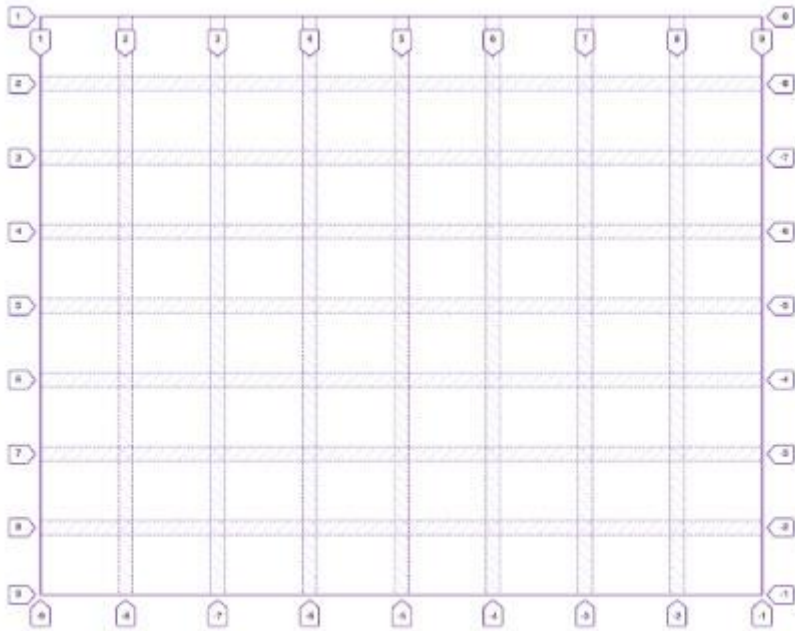
.okno4 {
  grid-column-start: 1;
  grid-column-end: 5;
  grid-row-start: 3;
  grid-row-end: 6;
}

.okno5 {
  grid-column-start: 1;
  grid-column-end: 5;
  grid-row-start: 6;
  grid-row-end: 9;
}

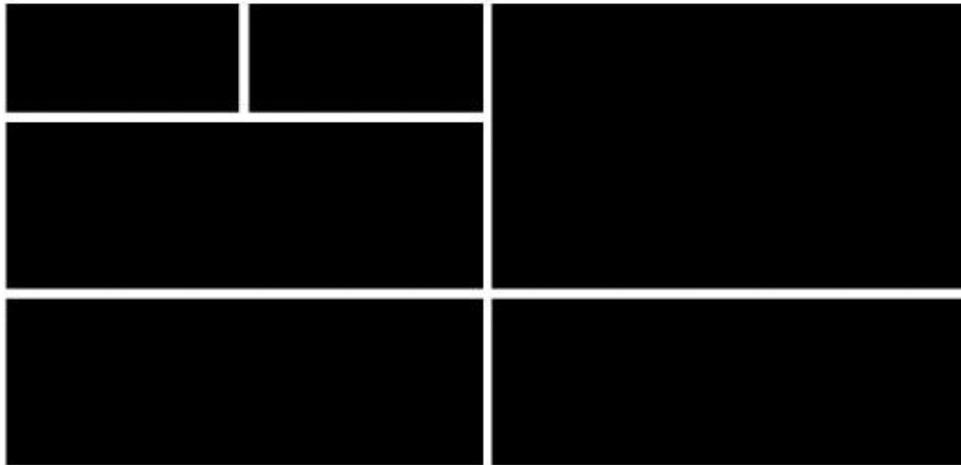
.okno6 {
  grid-column-start: 5;
  grid-column-end: 9;
  grid-row-start: 6;
  grid-row-end: 9;
}
```

- **grid-column-start: 1;** – nastav aby element okno1 začínal v stĺpci 1
- **grid-column-end: 3;** – nastav aby element okno1 končil v stĺpci 3
- **grid-row-start: 1;** – nastav aby element okno1 začínal v riadku 1
- **grid-row-end: 3;** – nastav aby element okno1 končil v riadku 3

takto nastavím jednotlivé elementy okno1, okno2 .... okno6 tam kam chcem, samozrejme musím mať určitú predstavu čo znamená mriežka 8 x 8, pre lepšie pochopenie:



a výsledok voala:



gratulujem vytvoril si si svoj prvý CSS GRID layout a ani to nebolelo

**! v tomto prípade som nedefinoval žiadne grid-area a to z toho dôvodu že všetci priami potomkovia mriežky (to jest elementy okna) sa automaticky stávajú položkami (grid-area -mi mriežky.) ! – v nasledujúcom štýle zobrazenia gridu už ale budeme jednotlivé okná definovať ako grid-area**

## DRUHÝ spôsob

vytvorím si html:

```
<!DOCTYPE html>
<html lang="sk">
<head>
  <meta charset="utf-8">
  <title>Grid</title>
  <link rel="stylesheet" href="../grid/grid.css">
</head>
<body>
  <div class="mriezka">
    <div class="okna okno1">
    </div>
    <div class="okna okno2">
    </div>
    <div class="okna okno3">
    </div>
  </div>
</body>
</html>
```

vložím css:

```
html {
  height: 100%;
}

body {
  height: 100%;
}

.mriezka {
  display: grid;
  height: 100%;
  grid-template-columns: 1fr;
  grid-template-rows: 10% 80% 10%;
  grid-template-areas: "navigacia" "telo" "paticka";
  grid-gap: 1vw;
}

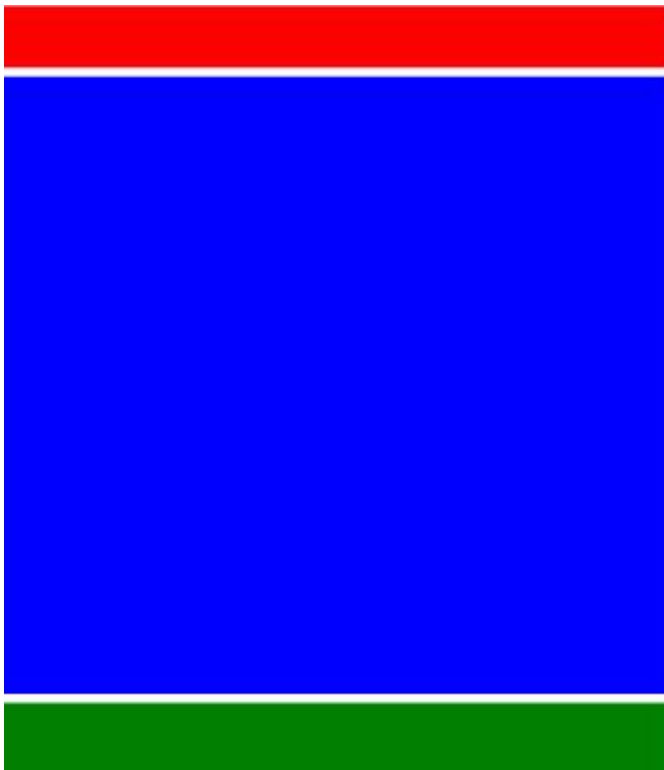
.okno1 {
  grid-area: navigacia;
  background-color: red;
}

.okno2 {
  grid-area: telo;
  background-color: blue;
}

.okno3 {
  grid-area: paticka;
  background-color: green;
}
```

- na to aby som mohol pracovať s % musím definovať výšku (height: 100%;) pre celý dokument t.j. : **html** , **body** aj **mriežka**
- **display: grid;** – už poznáš
- **grid-template-columns: 1fr;** – rozhodol som sa, že celú stránku budem považovať za 1 stĺpec ktorý budem deliť na jednotlivé horizontálne časti
- **grid-template-rows: 10% 80% 10%;** – rozhodol som sa, že navigácia to jest okno1 má mať výšku 10% celej stránky, telo t.j. okno2 stránky kam budem dávať môj obsah ( texty, obrázky...) má mať výšku 80% celej stránky a pätička t.j. okno3 bude zaberáť 10% výšky dokumentu.
- **grid-template-areas: "navigacia" "telo" "paticka";** – rozloženie bude úplne hore navigacia , v strede telo a na spodku paticka **!!! zatiaľ nemám zadefinované grid-area navigacia telo ani paticka – tie budem definovať jednotlivo !!!**
- **grid-template-areas** zadávam do ""
- **grid-area: navigacia;** / **grid-area: telo;** / **grid-area: paticka;** – sú jednotlivé časti mojej mriežky , pomocou príkazu grid-area som zadefinoval ktorá má byť ako pomenovaná (môžem pomenovať ako len chcem len nesmiem zabudnúť uviesť presný názov do grid-template-areas

a výsledok voala – pre lepšie znázornenie som pridal farby jednotlivým elementom:

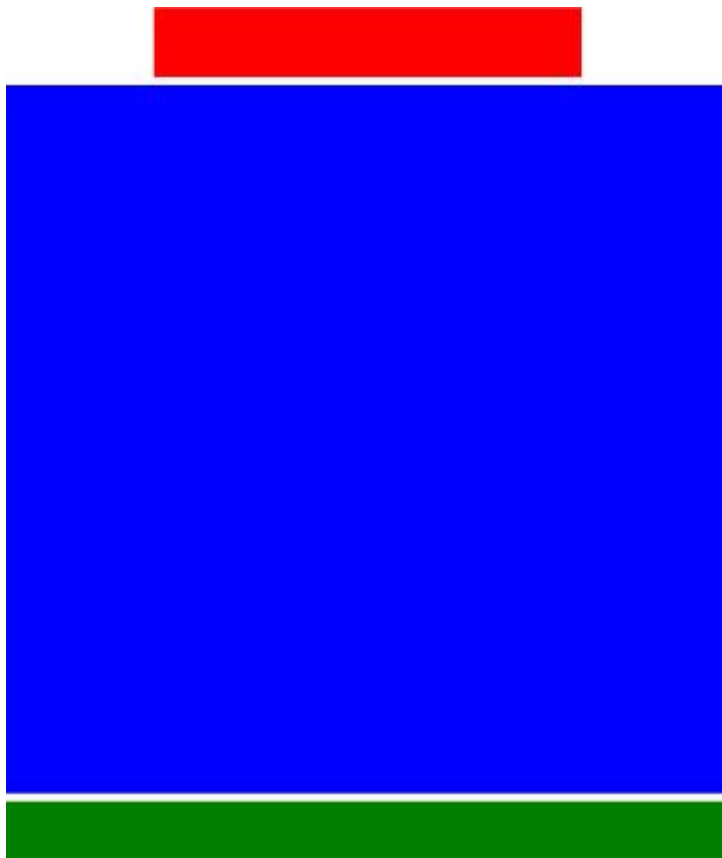


zoberme si príklad, že nechcem aby hlavička zaberala celú šírku stránky, je to možné?  
samozrejme áno malinkou zmenou css:

```
.mriezka {  
  display: grid;  
  height: 100%;  
  grid-template-columns: 1fr 3fr 1fr;  
  grid-template-rows: 10% 80% 10%;  
  grid-template-areas: ". navigacia. " "telo telo telo"  
    "paticka paticka paticka";  
  grid-gap: 1vw;  
}
```

- **grid-template-columns:** 1fr 3fr 1fr; – zmena oproti layoutu je tá, že celú stránku som rozdelil na 3 stĺpce (!s rozličnou šírkou)
- **grid-template-areas:** ". navigacia. " "telo telo telo" "paticka paticka paticka"; – ako môžeš vidieť keďže mám 3 stĺpce ktoré som si zadefinoval musím aj každý element definovať 3krát !!! ". navigacia ." znamená že navigácia má zaberáť len stredný stĺpec, tam kde sú bodky znamená, že tam nič nebude!!!

výsledok:



## ZÁVER:

Musím povedať, že CSS GRID je neuveriteľne mocný nástroj moderného webdizajnu, ktorý ti jednak zjednoduší tvoju prácu s umiestňovaním jednotlivých elementov na tie miesta ktoré chceš a to hlavné že ti zjednoduší a skráti kód potrebný k dosiahnutiu vytýčeného cieľa.

pre maximálny efekt odporúčam používať veľkosti : vw (view width) alebo %, predsa len chceme byť moderní a responzívny webdizajn je v dnešnej dobe nutnosť.

Ďakujem za pozornosť a snád' som ťa niečo naučil.