

---

# Zadání soutěžních úloh

## Kategorie tvorba webu

27. června 2015

Soutěž v programování – 28. ročník

Finále

Úlohy můžete řešit v libovolném pořadí a samozřejmě je nemusíte vyřešit všechny. Počet bodů za každou úlohu je uveden přímo v jejím zadání. Hodnotí se shoda se zadáním, funkčnost, dodržování webových standardů a přehlednost zdrojového kódu.

Před zahájením soutěže vám pořadatel oznámí, kde najdete testovací soubory a kam máte ukládat řešení úloh. Všechna vaše řešení ukládejte do adresáře `wnn`, kde `nn` je vaše startovní číslo. Kompletní řešení každé úlohy (soubory HTML, CSS, obrázky, případně Javascript) uložte do samostatného podadresáře nazvaného jménem úlohy (např. `kalendar`). Stránku s řešením pojmenujte `index.html` (pokud ze zadání nevyplývá jinak) a uložte v kódování UTF-8.

Při zápisu HTML a CSS kódu dodržujte webové standardy. Součástí hodnocení vašich řešení je i kontrola zdrojových kódů pomocí validátorů. Doporučujeme používat HTML5, ale můžete použít i starší verze – HTML4 nebo XHTML1.

Při řešení je povoleno používat knihovnu jQuery bez pluginů. Další obdobné frameworky pouze po schválení porotou.

# Menu

max. 10 bodů

Vytvořte soubor `style.css`, který definuje CSS styl k dodanému souboru `aplikace.html`. Do souboru `aplikace.html` nesmíte nijak zasahovat.

Po nastylování se stránka zobrazí jako návrh webové aplikace, kde se u horního okraje zobrazí menu a zbytek stránky obsahuje text.

Menu je standardně sbaleno do jediného řádku těsně u horního okraje stránky, viditelná jsou tedy vedle sebe slova **Soubor**, **Úpravy**, **Zobrazení**, **Nástroje** a **Nápověda**. Mezi slovy je mezera o šířce 2-3 znaků. Celý řádek je podbarvený světle šedou barvou, mezi tímto šedým pruhem a okraji stránky nahoře a po stranách není žádná mezera. Tomuto řádku budeme dále říkat *hlavní menu*.

Všechny texty hlavního menu i všech podmenu jsou černé, nepodtržené.

Po najetí myši na některou z položek hlavního menu se příslušná položka podbarví světle modře a pod ní se rozbalí podmenu. Dbejte přitom, aby při zobrazení podmenu zůstaly položky hlavního menu na svých místech. Levý okraj textu podmenu je zarovnaný s levým okrajem textu příslušné položky hlavního menu.

Podmenu je orámováno tmavě šedou barvou, pozadí podmenu je šedé, světlejší než pozadí hlavního menu. Při najetí myši na položku podmenu se tato položka podbarví světle modře (v celé šířce podmenu).

Obsahuje-li podmenu další podmenu, jsou tato další podmenu při zobrazení sbalená (skrytá). Při najetí myši na příslušnou položku je podmenu zobrazeno napravo od stávajícího podmenu. Totéž platí, pokud i toto podmenu obsahuje další podmenu.

Aktivní oblastí položky podmenu (tedy oblastí, na kterou lze kliknout, příp. oblastí, která při přejetí myši zobrazí podmenu) je celá šířka podmenu. Jinými slovy, k rozbalení podmenu se seznamem tiskáren musí stačit najet myši někam vpravo od slova **Tisk**.

Některá podmenu obsahují oddělovače (element `li` se třídou `sep`). Oddělovače jsou zobrazeny jako tmavě šedá vodorovná čára mezi položkami podmenu. Vzdálenost oddělovače musí být stejná od položky nad ním i pod ním.

Položky podmenu (nikoliv hlavního menu), které obsahují další podmenu, by měly tuto možnost vyznačit šipkou vpravo u pravého okraje podmenu.

Při skrolování textu by mělo menu zůstat stále viditelné při horním okraji obrazovky.

Při tisku stránky se menu nevytiskne.

Funkčnost zbytku stránky by měla být zachována, tedy odkazy fungují a jsou odlišitelné od textu, odstavce jsou od sebe odděleny.

# Tabulky

max. 20 bodů

V Javascriptu vytvořte knihovnu, která umí generovat interaktivní HTML tabulky. Knihovna bude uložena v samostatném souboru *tables.js*.

API knihovny je zadáno u popisu požadavků.

## Postup práce s tabulkou:

1. Nastavení vlastností
2. Nastavení dat
3. Zobrazení
4. Interakce

### Nastavení vlastností

Tabulka by měla umožnit nastavení následujících hodnot:

- Nadpis – zobrazuje se nad tabulkou, je zarovnaný na střed stránky

```
setTitle(title)
```

- Počet sloupců

```
setColumnCount(n)
```

- Nastavení hlaviček sloupců – tedy co se zobrazuje v hlavičkách tabulky

```
setHeaderTitles(headerNames)
```

- (Maximální) počet zobrazovaných řádků

```
setRowCount(n)
```

### Nastavení dat

Data by se do tabulky měla dát přidávat následujícími způsoby:

- Po řádcích – pomocí metody, do které se vkládá stejný počet parametrů jako je sloupců v tabulce

```
addRow()
```

- Po řádcích – vstupem metody je řetězec, který obsahuje hodnoty sloupců oddělené středníkem. V případě špatného vstupu knihovna zobrazí chybovou hlášku a řádek se nenačte.

```
addRowFromString(rowString)
```

Příklad:

```
leden;-2.4;4.7
```

- Hromadně – data jsou předána pomocí JSON ve formátu pole řádků, kdy řádek je pole hodnot. Například hodnota

```
[
    ["leden", "-2.4", "4.7"],
    ["únor", "-1.4", "3.5"],
    ["březen", "3.4", "8.0"],
    ["duben", "8.4", "14.0"]
]
```

obsahuje 4 řádky tabulky, každý řádek obsahuje 3 hodnoty. Pokud při načítání dat přijdou na vstup špatně zformátovaná data (tedy např. se špatným počtem sloupců), zobrazí se chybová hláška s informací o chybě a daty, které chybu způsobily.

```
addRowsJson(json)
```

## Zobrazení

Příkaz, který způsobí vygenerování HTML, místo vložení HTML si uživatel volí pomocí id prvku.

```
createTable(id)
```

## Interakce

Tabulka by měla mít následující chování:

- Když se kurzor nachází nad libovolnou buňkou řádku kromě hlavičky, je tento řádek podbarven jinou barvou než zbytek tabulky.
- Když se kurzor nachází nad hlavičkou některého sloupce, je celý sloupec podbarven jinou barvou než zbytek tabulky.
- Při kliknutí na hlavičku sloupce jsou řádky tabulky zobrazeny seřazené podle hodnot v tomto sloupci. V hlavičce sloupce se zobrazí šipka indikující směr seřazení.
- Při kliknutí na hlavičku již seřazeného sloupce se směr třídění otočí. Šipka v hlavičce indikuje nový směr třídění.
- Při dvojkliku na libovolnou datovou buňku se v buňce zobrazí input s předvyplněnou hodnotou buňky. Hodnota v inputu se dá měnit a po stisknutí klávesy Enter se hodnota buňky změní na zadanou hodnotu v inputu. (pokud byly řádky seřazené podle tohoto sloupce, tak se může stát, že po změně už seřazené nebudou).

Zvolte vhodné barvy.

## Příklad

Sekvence příkazů:

```
tables.setTitle("teploty");
tables.setColumnCount(3);
tables.setHeaderTitles(['month', 'min temp', 'max temp'] );
tables.addRow("leden", "-2.4", "4.7");
tables.addRow("únor", "-1.4", "3.5");
tables.addRow("březen", "3.4", "8.0");
tables.addRow("duben", "8.4", "14.0");
tables.addRowFromString("květen;9.9;15.2");
```

```
tempsSummer = [  
    ['červen', '12.4', '19.0'],  
    ['červenec', '17.4', '25.2']  
];  
tables.addRowJson(tempsSummer);  
tables.setRowCount(6);  
tables.createTable("here");
```

vytvoří tabulku:

teploty

month	min temp	max temp
leden	-2.4	4.7
únor	-1.4	3.5
březen	3.4	8.0
duben	8.4	14.0
květen	9.9	15.2
červen	12.4	19.0

(zobrazují se jenom 6 řádků kvůli volání funkce *setRowCount(6)*).

## Galerie

*max. 30 bodů*

Vytvořte webovou galerii obrázků. Obrázky a jejich popis získáte ze serveru 192.168.2.222, dokumentaci API naleznete na konci úlohy.

Galerii můžete vytvořit, aniž by stránka se serverem komunikovala, tedy příslušná data budete mít uložena přímo v HTML souboru, ale dostanete méně bodů. Stejně tak nemusíte splnit všechny body popsané níže (např. nemusíte podporovat přepínání fotoalb), ale dostanete adekvátně méně bodů.

Při načtení stránky bude zobrazeno libovolné z fotoalb.

Na horním okraji stránky se nachází rozbalovací pole formuláře se seznamem dostupných fotoalb. Výběr fotoalba ze seznamu způsobí jeho zobrazení.

Název fotoalba je napsán svisle u levého okraje stránky, text je otočený o 90° proti směru hodinových ručiček. Náhledy fotek a název fotoalba překrývat.

Vedle názvu alba se zobrazují náhledy fotek. Náhled každé fotky má výšku 200 pixelů, jeho šířka je taková, aby byl zachován poměr stran původního obrázku. Náhledy jsou zobrazeny po řádcích vedle sebe až do maximálního možného zaplnění šířky okna, poté pokračují na dalším řádku, atd. Mezi fotkami je úzká mezera, stejně velká vodorovně i svisle. Při změně šířky okna se náhledy přeskládají, aby předchozí stále platilo.

Při najetí myši na fotku se v její spodní části objeví její popisek. Popisek se nachází na pruhu, který zabírá celou šířku fotky, a který zesvětluje fotku pod pruhem tak, aby popisek byl čitelný. Popisek je vždy jednořádkový, pokud se na fotku nevejde, ořízněte ho. Pokud fotka popisek nemá, při najetí myši se pruh nezobrazí.

Po kliknutí na fotku se zobrazí plná velikost fotky (případně, pokud chcete, velikost fotky omezená velikostí obrazovky). Pokud má fotka popisek, je zobrazen pod fotkou.

Při najetí na plnou velikost fotky se u jejího pravého a levého okraje objeví šipky. Kliknutí na šipky zobrazí následující nebo předchozí fotku. Kliknutí mimo fotku (případně, pokud chcete, na zavírací ikonku, kterou někam vhodně umístíte) zavře zobrazení plné velikosti fotky a opět zpřístupní náhledy. V každém případě musí jít fotka nějakým způsobem zavřít (pozor na to zejména, pokud vám fotka zabírá celou velikost okna).

Při zobrazení plné velikosti fotky lze stiskem kláves (na klávesnici) doprava a doleva fotky posouvat, klávesou `Esc` fotka zavřít, klávesou `F` přepnout do režimu přes celou obrazovku (full screen).

Kliknutím a tažením náhledu fotky je možno měnit pořadí fotek. Procházení fotek v plné velikosti respektuje aktuální pořadí fotek. Tato část úlohy bude hodnocena větším počtem bodů.

Za pěkné grafické zpracování celé úlohy můžete získat body navíc.

## Popis API

URL pro komunikaci se serverem má tvar: `http://192.168.2.222/api/formát/identifikátor`

*formát* je buď `json`, nebo `txt`. Formát `txt` je čitelný pro lidi a je vhodný, pokud se rozhodnete splnit úlohu jen částečně a data vložit přímo do HTML stránky. Formát `json` je strojově čitelný a je vhodný pro komunikaci vaší stránky se serverem (nic vám však nebrání použít i pro tento účel formát `txt`, chcete-li).

*identifikátor* udává, jaká data po serveru chcete. Pro získání seznamu fotoalb slouží identifikátor `index`. Seznam fotoalb pak obsahuje identifikátory jednotlivých fotoalb.

Všechny texty jsou v kódování UTF-8. Konce řádků jsou CRLF.

### Formát seznamu fotoalb (index) – json

Server vrátí pole. Každý prvek pole je jedno fotoalbum. Fotoalbum je objekt s následujícími prvky (properties):

`id`

Identifikátor alba. Pro načtení obsahu fotoalba použijte tuto hodnotu jako *identifikátor* v URL uvedeném výše.

`name`

Název fotoalba.

### Formát obsahu fotoalba – json

Server vrátí pole. Každý prvek pole představuje jednu fotku. Fotka je objekt s následujícími prvky (properties):

`url`

Adresa fotky. Vždy jde o platné URL. Můžete předpokládat, že fotka je ve formátu JPG a není větší než 1024 x 1024 pixelů.

`description`

Popisek fotky. Tento prvek nemusí být přítomen, v takovém případě fotka popisek nemá.

### Formát txt

U seznamu fotek je to název alba, URL s daty o albu.

U dat o albu je to popisek, URL fotky.

## Příklad použití API

Získání seznamu všech alb v lidsky čitelné podobě: <http://192.168.2.222/api/txt/index>

Získání alba s identifikátorem budovy ve formátu json: <http://192.168.2.222/api/json/budovy>