



ŘEŠENÍ

1 Barevný generátor

2. barva: **#00B86B** - V modelu RGB bylo zapotřebí změnit složku "R" z hodnoty 189 na 0 - odstranění červené barvy
3. barva: **#006A4E** - V modelu CMYK přičíst ke složce "M" a "K" hodnoty 20
4. barva: **#4E6A00** - V modelu RGB změnit hodnoty "R" na 78 a "B" na 0 - tudíž prohození hodnot "R" a "B"
5. barva: **#B195FF** - V modelu RGB invertovat všechny složky. Inverzi lze jednoduše provést odečítáním jednotlivých složek od maximální hodnoty 255

6. barva: **#6C6580** - V modelu HSV odečtením z "S" a "V" 50% své hodnoty - "21" a "50"

Výsledná barva: HEXA: **#6C6580**; RGB: **108 101 128**; CMYK: **16 21 0 50**

Při bodování byl zohledněn zejména postup při počítání jednotlivých kroků. Pokud se výsledná barva nepatrně lišila, byl přidělen 1 bod, pokud byla shodná s výsledkovou listinou, byly uděleny 2 body. Nejčastější chyby byly ve 2. kroku. Místo k *přičtení* k výchozí barvě hodnoty 0 20 0 20 došlo k *přepsání* na tyto hodnoty. Vyšla barva, která i po inverzi v modelu RGB byla naprosto stejná a při pokračování v dalších krocích docházelo k absolutnímu znehodnocení barvy. Tyto dva kroky měly napovědět, že tento postup není správný nebo že došlo k chybě.

2 Programování – tutor.fi.muni.cz

Na 4 body – s použitím opakování = repeat:

Hvězda: repeat 8 [fd 50 bk 50 rt 45]

Schody: repeat 3 [rt 90 fd 50 rt 90 fd 50 lt 90 lt 90]

Kříž: repeat 2 [fd 150 rt 90 fd 50 rt 90 fd 150 rt 90 fd 50 rt 90 fd 50 rt 90
penup fd 100 rt 180 pendown]

3 Přřadit k sobě čísla v různých číselných soustavách

1. Trojice čísel: **2-2-10** (desítková-šestnáctková-dvojková)
2. Trojice čísel: **19-13-10011** (desítková-šestnáctková-dvojková)
3. Trojice čísel: **15-F-1111** (desítková-šestnáctková-dvojková)
4. Trojice čísel: **16-10-10000** (desítková-šestnáctková-dvojková)

Vyluštěná písmena: **R, B, O, N**

Slovo: **BRNO**

4 Šifra

Nápověda: klíč: +- 223. Nad každým písmenem zašifrované zprávy je 2 nebo 3, + a – znamená, že se máte v abecedě posunout o 2 nebo 3 písmena, Q-2=O, U-2=S, V-3=S a tak dále.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2	2		3		2	2	3	2	2		3	2	2	3	2	2	3								
Q	U		V		N	Q	J	G	O		W	W	E	Q	C	M	D								
O	S		S		L	O	G	E	M		T	U	C	N	A	K	A								

je **L I N U X**

5 Levely

Každý level šel splnit dvěma způsoby: První bylo zadání "hesla" do formuláře, druhým bylo zadání

hesla přímo do URL adresy (toto řešení vyžadoval level 5). Úloha byla spíše o logickém myšlení než o znalostech, které ovšem rozhodně nebyly na škodu.

Level 1:

Ve zdrojovém kódu bylo: <title>Purkiáda 2013 - **h4Ck T3\$t**</title>. Stačilo rozluštit **1337 \$p34K** (<http://cs.wikipedia.org/wiki/Leetspeak>) – jazyk nebo způsob primitivního kryptování písma na internetu běžně používaný. Těžší znaky zde byly zobrazeny normálně. Heslo: **leetspeakrullez**

Level 2:

Level, který obsahoval známý mem s nápisem: "We need to go deeper". Tato nápověda vás měla nasměřovat hlouběji do stránky, resp do jejího zdrojového kódu (nápověda z předchozího levelu s CTRL+U). Ve zdrojovém kódu stránky je zde oproti prvnímu levelu přidán javascriptový kód check(), který porovnává obsah formuláře s proměnou "a" a případně přesměruje do dalšího levelu, nebo vás vyzve aby jste to zkusili znovu. Heslo: **null**

Level 3:

Asi nejtěžší level. Nápovědný obrázek zobrazuje jpg soubor otevřený v textovém editoru. Stačilo tedy nápovědný obrázek (jiný obrázek se na stránce v bloku s hrou nenachází) stáhnout a otevřít např. v poznámkovém bloku. Aby se hledání hesla urychlilo, stačilo použít hledací funkci a napsat: heslo, popřípadě se podívat na meta-data obrázku, kde se heslo zobrazilo taky.. Heslo: **cheese_pizza**

Level a87ff679a2f3e71d9181a67b7542122c (4):

Název levelu (a87ff679a2f3e71d9181a67b7542122c) napovídá, že heslo bude nějak zašifrované. Heslo v zahashované podobě (MD5 hash chcete-li) se zobrazí opět ve zdrojovém kódu stránky.

```
<!-- Heslo? Jasný, to je přece c98d8230ab2f948422a219f1eca33658 !-->
```

Hash je sice jednosměrný, ale dnes existují databáze slovníkových hashů. A kde jinde hledat než na googlu. Po zadání hesla do googlu mi vyjedou dva odkazy, přičemž hned v tom prvním mě praští do očí nápis: "hash result is: Diplodocus". Heslo: **Diplodocus**

Level 5:

Tento level prověřoval vaši pozornost. Oproti ostatním levelům zdrojový kód obsahuje navíc jeden .css soubor. V tomto souboru jsou napsána hesla ke dvěma předešlým levelům, takže je jasné, že třetí hodnota bude nejspíše heslo k tomuto. Tuto hodnotu stačilo zadat ve kterémkoliv levelu předcházejícím (kromě levelu 2) nebo ji dát jako url parametr stránky ((hrle= hacked_101, ti všímavější si určitě všimli, že vždy obsahuje předcházející heslo). Heslo: **hacked_101**

6 GPS šifra

Nejlehčí úkol Purkiády je vlastně ještě lehčí, než vypadá. Hlavním úkolem je jen vyznačit a spojit dané souřadnice v mapě, z nichž vyjde šifra. Jde to několika způsoby, například primitivně hledat souřadnice v mapě a kreslit spojení bodů, anebo je jednoduše zadat do plánovače tras. Ideální je tedy využít Google Maps s jeho vyhledáváním tras. Obvyklému vyhledání trasy stačí počáteční a koncový bod, ale pro naši šifru si musíme přidat dalších **16 cílů**, přes které trasa povede. Výchozím nastavením je plánování trasy pro auto, jenže v místech naší šifry se nachází jednosměrné komunikace. Proto musíme nastavit plánování pro pěší, jinak nám to naplánuje trasu přes nejbližší objížďky a šifra bude absolutně zkreslená. Dáme vyhledat trasu a v mapě se vám zobrazí zakreslené logo známé společnosti **ASUS**. Nakonec jen zapíšete 3 horizontálně položené ulice, po kterých trasa vede. A to jsou "**East Lafayette Ave**", "**East Lanvale Street**" a "**East Federal Street**".

7 Osmisměrka

Tajenka: **TABULKOVÝ EDITOR MICROSOFTU** (největší problém bylo vypsat zbylá písmena, někteří tady i tuto oddechovku vzdali rovnou). Odpověď: **(MS) EXCEL (2007, 2010)** (uznávali jsme odpovědi, které se aspoň podobaly – calc ovšem ne).