Řetězce (kódování znaků, regulární výrazy, funkce pro práci s řetězci)

Řetězce

Řetězec je datový typ sloužící k uložení posloupnosti znaků. Řetězec může být *konstantní* (obsah proměnné nelze měnit), se *staticky alokovaným prostorem* (řetězec má omezenou maximální délku) nebo s *dynamicky alokovaným prostorem* (řetězec má maximální délku omezenu jen velikostí volné paměti).

Kódování znaků

Kódování znaků je způsob prezentování binárně zapsaných znaků v aplikaci či např. operačním systému. Existuje velká spousta různých kódování, dnes je nejpoužívanější UTF-8, které umí zapsat znaky z různých národních abeced. Pro češtinu se dříve používalo například windows-1250 či iso-8851-2.

V Číně či např. Japonsku se používá kódování UTF-16, které obsahuje znaky národních abeced zmíněných zemí, UTF-32 pak zahrnuje také znaky již zaniklých abeced a jazyků, jako byla např. hlaholice. Číslo za pomlčkou určuje, na kolika bajtech je zapsán jeden znak. Kódování UTF jsou částmi tzv. tabulky UNICODE.

Za prapůvodní kódování lze považovat ASCII tabulku, která nejprve obsahovala 128 znaků, a to anglickou abecedu, číslice a speciální znaky. Později byla rozšířena na 256 znaků a zahrnovala také znaky některých národních abeced. Prvních 128/256 znaků všech kódování obvykle odpovídá ASCII tabulce.

Regulární výrazy

Regulární výraz je speciálně zapsaný řetězec, který definuje tvar jiného řetězce. Používá se pro kontrolu, zda je nějaký řetězec v požadovaném tvaru (například PSČ či nějaké datum). Rozlišují se dva typy zápisu regulárního výrazu – tzv. *posixový* a *perlový*.

Znak	**Význam**	**Příklad**	
|	alternativa	ano|ne	„ano“, nebo „ne“
(a)	substituce za dílčí reg. výraz	dom(ov|ecek)	„domov“ nebo „domeček“
[…]	jeden z uvedených zápisů	[a-z0-9]	a až z nebo 0 až 9
[^…]	jeden z neuvedených zápisů	[^a-z0-9]	vše kromě a až z nebo 0 až 9
.	libovolný znak	.olo	„kolo“, „molo“, …
^…	pouze na začátku řetězce	^ahoj	„ahoj Honzo“, „ahojahoj“, …
…$	pouze na konci řetězce	tratit$	„ztratit“, „utratit“, „vytratit“, …
?	nepovinná část	xy?	„x“ nebo „xy“
*	libovolný počet opakování	x*	„“, „x“, „xx“, „xxx“, „xxxx“, …
+	lib. počet opak., ale min. 1	x+	„x“, „xx“, „xxx“, „xxxx“, …
{…}	počet opakování	x{3,5}	„xxx“, „xxxx“, nebo „xxxxx“

Některé typy znaků mohou být reprezentovány určitým písmenem. Například desítkové číslice \d, bílé znaky pak \s.

`\d{3} \d{2}`	české PSČ
`[0-9a-fA-F]+(, ?[0-9a-fA-F]+)*`	seznam HEX čísel oddělených čárkou a nepov. mezerou

Funkce pro práci s řetězci

S řetězci je, zejména ve webových aplikacích, potřeba neustále pracovat. Proto existují funkce, které takovou práci s řetězci umožňují. Mezi nejpoužívanější funkce patří např. `strlen()`, která vrací délku řetězce – resp. `mb_strlen()`, která umožňuje druhým argumentem určit kódování, `trim()` ořezávající bílé znaky ze začátku a z konce znaku, `str_replace()` pro nahrazení nějaké části řetězce nějakým jiným řetězcem či například `html_special_chars()` nahrazující speciální znaky v HTML (např. „<“ či „&“) speciálními entitami, např. < či & Další funkcí pro práci s řetězcem je také `strrev()`, která řetězec obrací.

Dalšími velice používanými funkcemi jsou také `substr()` pro vybrání substringu ze stringu nebo funkce ověřující správnost/tvar řetězce v nějaké proměnné: `preg_match()`.