

# Datové typy (statické a dynamické typování, vznik proměnné, uložení dat v paměti, práce s primitivními a neprimitivními datovými typy, generické typy)

## Datové typy obecně

Datový typ určuje druh hodnoty, která je uložena v proměnné. Mezi datové typy patří například *string* (řetězec), *integer* (celé číslo), *float* (číslo s plovoucí desetinnou čárkou), *boolean* (true/false).

## Statické typování

Při statickém typování se při deklaraci proměnné určí, pro jaký datový typ je proměnná určena. Hodnota proměnné je pak omezena na datový typ určený při její deklaraci. Používá se například v jazycích Java, C++ či C#.

Výhodou je například známost typu při kompilaci a možnost kontroly kódu – zda se například nesnažíme přičíst číslo ke stringu – kompilátor tak přijde na chybu a kód nezkompiluje – dokud nebude odladěn. Aplikace je také ve výsledku rychlejší, výkonnější.

Nevýhodou je, že proměnná může nést hodnotu pouze určeného datového typu a obcházení tohoto omezení dělá program složitějším.

## Dynamické typování

Datový typ je při dynamickém typování sdružen s hodnotou. Uložíme-li do proměnné hodnotu jiného datového typu, změní se datový typ celé takové proměnné. Mezi příklady patří PHP, Python či například JavaScript.

Mezi výhody patří flexibilita a snadnost při programování – možnost uložit do proměnné kdykoliv jakýkoliv datový typ a za běhu programu jej libovolně modifikovat.

Mezi nevýhody patří horší optimalizace výkonu – programy jsou většinou pomalejší a chyby v datových typech jsou odhaleny až při běhu programu.

## Vznik proměnné a uložení dat v paměti

Proměnná vzniká svojí deklarací. Součástí deklarace může být již zmíněné určení datového typu deklarované proměnné. Při vzniku proměnné je pro ni vyhrazen prostor v paměti – právě podle datového typu. U dynamického typování se vyhrazená velikost v paměti může měnit (je-li třeba).

PHP: 

```
$variable = "string hodnota";
$another = false;
```

Java: 

```
String variable = "string hodnota";
boolean another = false;
```

JavaScript: 

```
var variable = "string hodnota";
var another = false;
```

## Uložení dat v paměti

Každý datový typ v paměti zabírá jinou velikost.

| Datový typ                   | Délka   | Rozsah hodnot   |
|------------------------------|---------|---|
| boolean                      | 1 bit   | 0 až 1  |
| byte/unsigned char           | 1 bajt  | 0 až 255  |
| word/unsigned short int      | 2 bajty | 0 až 65 535   |
| long word/unsigned long int  | 4 bajty | 0 až 4 294 967 295                                      |
| integer/long int             | 4 bajty | -2 147 483 648 až 2 147 483 647                         |
| double integer/long long int | 8 bajtů | -9 223 372 036 854 775 808 až 9 223 372 036 854 775 807 |
| real/float                   | 4 bajty | 1E-37 až 1E+37 (6 desetinných míst)                     |
| double float/double          | 8 bajtů | 1E-307 až 1E+308 (15 desetinných míst)                  |

## Práce s primitivními a neprimitivními datovými typy

Práci s primitivními datovými typy většinou zvládá samotné jádro programovacího jazyka. Jde například o typy boolean, integer, float, char či enum. Primitivní datové typy vždy obsahují pouze jeden prvek (např. číslo, true/false, znak, nebo index pro enum).

Oproti tomu neprimitivní (nebo také složené) datové typy mohou obsahovat jeden nebo i více prvků – jedná se například o string (obsahuje několik znaků) či array. Práce se složenými datovými typy

## Generické typy

Generický datový typ je datový typ, který může zastupovat více datových typů. Genericitou se v některých programovacích jazycích řeší například nemožnost uložit do pole integerů řetězcovou hodnotu. V Javě se používá například u ArrayListu, což je ve své podstatě pole, do něhož můžeme uložit několik libovolných datových typů.