# Pole v C

## Statická pole

Pole je homogenní, tedy obsahuje v celém svém prostoru jen jeden datový typ. Třeba čísla.

```
int pole[10];
```

toto je pole 10 čísel. Pozor! Indexuje se od 0. Tedy pole[0] je první číslo a pole[9] je poslední číslo, pole[10] způsobí chybu, protože sáhnu mimo můj paměťový prostor.

```
char znaky[20];
```

Toto je pole znaků. Může obsahovat až 20 znaků. A toho využíváme při ukládání textů. Viz. Řetězce

## Dynamická pole

🔧Fix Me!

### Předpoklady

- Pointer
- Podmínky
- Konverze typů

### Dynamicky alokovaná paměť

Většinou v době překladu programu neznáme velikost našich dat (vlastně skoro nikdy). Proto potřebujeme za chodu programu (třeba po tom, co nám uživatel řekl, kolik položek bude zadávat) získat paměť pro data, která bude program ukládat.

A na to nám slouží funkce malloc(), calloc(), realloc() a free().

Onoseto dopíše: Co je to ten typ void*? Proč malloc() vrací ukazatel? Proč to musí být tolik psaní?

void* je podle mě ukazatel, který neříká, na jaký typ dat ukazuje, protože funkce malloc() neví, pro co alokuje data - je to pouze ukazatel do paměti (?? ale ruku do ohně bych za to nedal.. :D)

malloc() alokuje blok paměti o dané velikosti a vrací ukazatel void*, který ukazuje na její začátek. Je to funkce určená pouze k alokování paměti, né k psaní do paměti ani ničemu jinému - alokuje paměť a předá nám na ni ukazatel, se kterým můžeme dále pracovat (kdyby nám ukazatel nepředala, nevěděli bychom, kde se daný blok alokované paměti nachází a nemohli bychom do něj později zapisovat...)

From:
https://wiki.gml.cz/ - **GMLWiki**

Permanent link:
**https://wiki.gml.cz/strprg:c:pole**

Last update: **04. 02. 2014, 22.36**