Функции

Определение

- Именована част от кода, съдържаща поредица от команди, която може да бъде извикана многократно за изпълнение.
- При извикването на функцията може да и бъдат подадени набор от стойности, наречени нейни параметри.
- След изпълнението си тя може да върне резултат.

А казано по-простичко?

Функцията е инструмент – създадена е да върши нещо:

- извикваме я по име, подавайки и входни данни
- тя върши работата си, като изпълнява поредица от команди
- накрая връща резултат там, където сме я извикали

Деклариране на функция

function име(*списък формални параметри*)
{
  команди;
  return резултат;
}

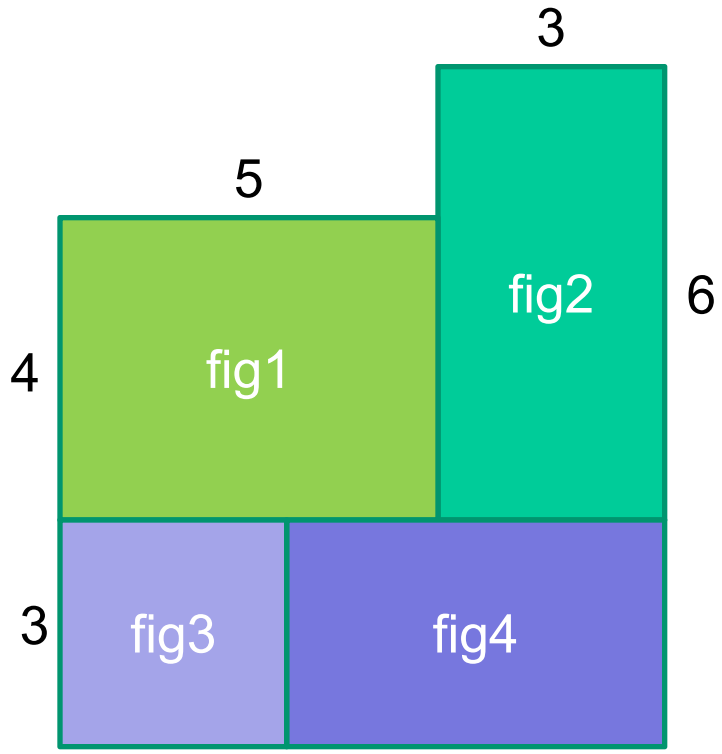
- име – името на функцията, чрез което ще я извикваме
- *списък формални параметри* – описват името на параметрите, които са необходими на функцията за да работи.
- *команди* – командите, които ще се изпълнят при извикването на функцията
- резултат от работата на функцията

Извикване на функцията

- име(*фактически параметри*);
  - име – името на функцията, която ще се изпълнява
  - фактически параметри – набор от стойности, които ѝ подаваме. Трябва да съответстват на формалните параметри.
- Действие: Изчисляват се фактическите параметри и се подават като стойности на съответните формални параметри на функцията. Тя се изпълнява. Връщаният от нея резултат се замества на мястото на извикването ѝ.

Пример за функции

```
function lice(a, b)  {
   return a*b;
}

var fig1 = lice(5, 4);
var fig2 = lice(3, 6);

alert( lice(5, 4) + lice(3, 6) +
          lice(3+5, 3) );
```



Предаване на параметри

Освен ако не е изрично указано, информацията необходима на функцията трябва **да ѝ се подаде като параметри**, а **НЕ да се въвежда във функцията**.

Неправилно:

```
function lice(a, b)  {
   a = prompt("a=");
   b = prompt("b=");
   return a*b;
}
```

Правилно:

```
function lice(a, b)  {
   return a*b;
}
```

Връщане на резултат

Освен ако не е изрично указано, функцията **връща резултат**, а **НЕ го отпечатва на екрана**.

Неправилно:

```
function lice(a, b)  {
   alert(a*b);
}
```

Правилно:

```
function lice(a, b)  {
   return a*b;
}
```

Връщане на резултат - особености

- трябва да е ясно какъв ще е резултатът на функцията за всеки един набор от входните параметри
- не трябва да има случай, в който не е указано какво ще върне функцията като резултат
- тогава тя пак ще върне стойност, но тя ще е недефинирана

Област на видимост

локални променливи

- декларирани някъде **във** функцията
- видими са **от началото до края на функцията**
- имат стойност **оттам, където им е указана**
- съществуват докато **функцията се изпълнява**

глобални променливи

- декларирани са **извън** всички функции
- видими **отвсякъде** (стига да не ги скриват локални променливи)
- имат стойност **оттам, където им е указана**
- съществуват докато **завърши изпълнението на програмата**

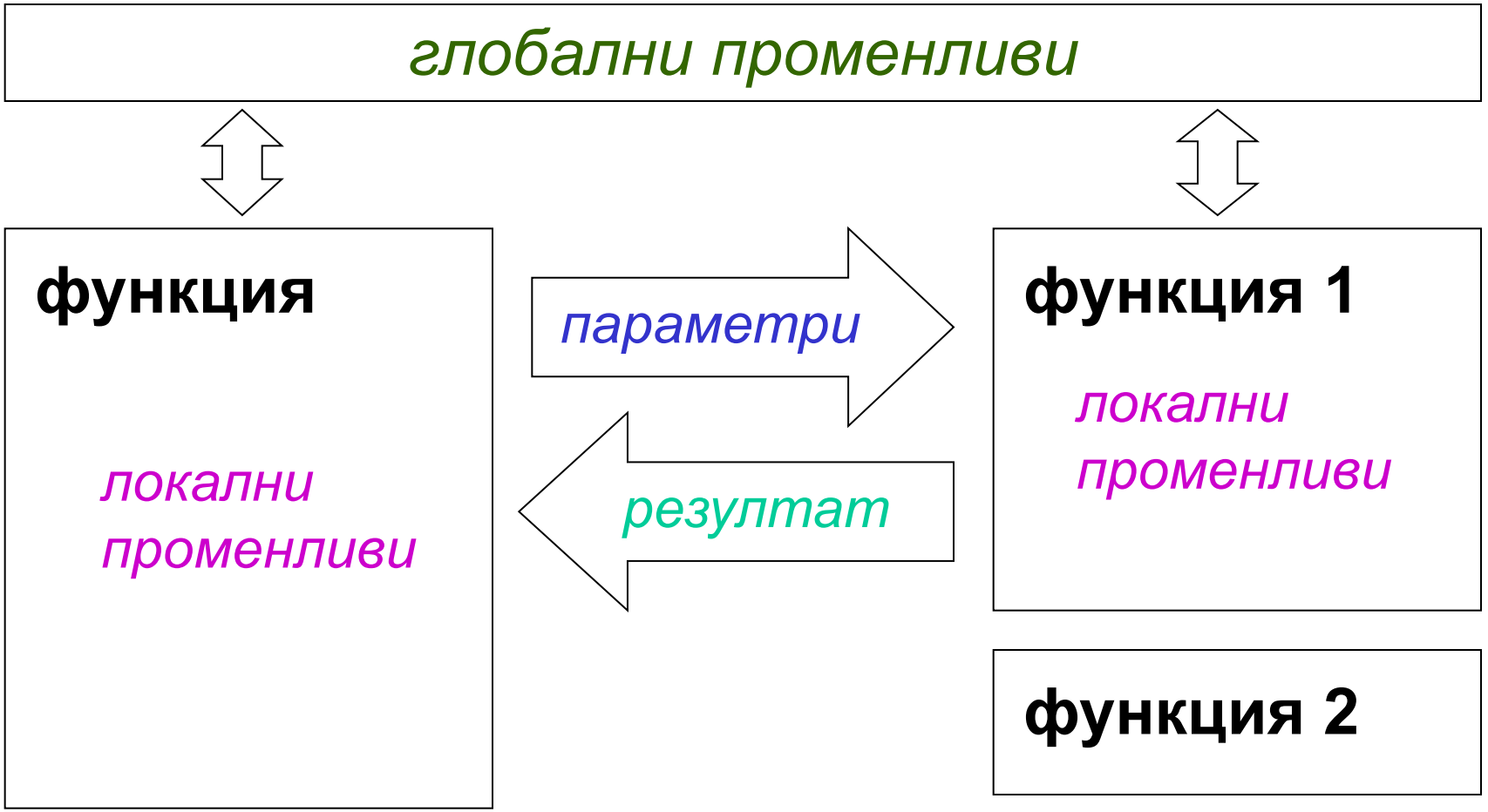
Пример: област на видимост

```
alert(a);     ----------------------------------------> undefined
var a = "global"; // глобална променлива

function testVar() {
   alert(a);  ----------------------------------------> undefined
   var a = "local"; // локална променлива
   alert(a);  ----------------------------------------> local
}

testVar();
alert(a);     ----------------------------------------> global
```

Кога кое да ползваме?

- Функцията е като инструмент – създадена е да върши нещо, и да върне резултат
- Като формални параметри и подаваме само онова, което е необходимо да знае, за да свърши работата си
- Ако за изчисленията и са нужни допълнителни величини, те се декларират в нея като локални променливи
- Глобални променливи декларираме, когато няколко функции ползват общи данни
- *Глобалните променливи трябва да се ползват пестеливо, защото правят функциите зависими една от друга*

Движение на данните



Край