Функции

(упражнения)

Вярно или невярно?

Напишете функция, която по дадени число X и интервал [A, B) връща дали числото X е в интервала [A, B).

```
function vInterval(x, a, b) {
    if ((x>=a) &&(x<b))
       alert("da“);
    else alert("ne“);
}

if (vInterval(4, 2, 10))
    alert(“В интерала е!“);
else alert(“Не е в интерала!“);
```

Невярно!

Напишете функция, която по дадени число X и интервал [A, B) връща дали числото X е в интервала [A, B).

```
function vInterval(x, a, b) {
    if ((x>=a) &&(x<b))
     alert("da“);
    else alert("ne“);
}

if (vInterval(4, 2, 10))
    alert(“В интерала е!“);
else alert(“Не е в интерала!“);
```

Тази функция не връща резултат, а го извежда на екрана!

Тук не е ясно какво е върнала функцията, защото в нея я няма командата **return**.

Вярно!

Напишете функция, която по дадени число X и интервал [A, B) връща дали числото X е в интервала [A, B).

```
function vInterval(x, a, b) {
    if ((x>=a) &&(x<b))
      return true;
    else return false;
}

if (vInterval(4, 2, 10))
    alert("В интерала е!");
else alert("Не е в интерала!");
```

Тази функция вече връща резултат…

… който можем да използваме в някакви други команди. Ако беше изведен на екрана, това нямаше да е възможно.

Ефективно!

Напишете функция, която по дадени число X и интервал [A, B) връща дали числото X е в интервала [A, B).

```
function vInterval(x, a, b) {
    return ((x>=a) &&(x<b));
}
```

Вместо „ако *изразът* е **верен** върни **true**, иначе върни **false**”, което беше на предния слайд, по-практично е директно да върнем *самия израз*, той така или иначе е **true** или **false**.

```
if (vInterval(4, 2, 10))
    alert(“В интерала е!“);
else alert(“Не е в интерала!“);
```

Вярно или невярно?

Напишете функция, която извежда по-голямото от две числа

```
function max(a, b) {
    if (a>=b)
       alert(a);
    else alert(b);
}

alert(max(2, 3));
```

Невярно!

Напишете функция, която извежда по-голямото от две числа

```
function max(a, b) {
    if (a>=b)
       alert(a);
    else alert(b);
}

alert(max(2, 3));
```

Тук не е ясно какво е върнала функцията, защото в нея я няма командата **return**.

Вярно!

Напишете функция, която извежда по-голямото от две числа

```
function max(a, b) {
    if (a>=b)
      alert(a);
    else alert(b);
}

max(2, 3);
```

Тъй като са ни поискали да направим функция, която директно **извежда** резултата…

… когато я извикваме просто трябва да напишем името ѝ и тя ще го изведе.

Вярно или невярно?

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( ) {
  var oc = prompt( “Въведи
                оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6))
     alert(“Не е оценка!“);
  else alert(oc);
}
```

Невярно!

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( ) {
  var oc = prompt( “Въведи
                оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6))
     alert(“Не е оценка!“);
  else alert(oc);
}
```

Къде тук виждате командата **return**, с която една функция казва колко ще върне като резултат?

Вярно или невярно?

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( oc ) {
  oc = prompt( “Въведи
              оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6))
     alert(“Не е оценка!“);
  else alert(oc);
}
```

Невярно!

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( oc ) {
  oc = prompt( “Въведи
              оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6))
     alert(“Не е оценка!“);
  else alert(oc);
}
```

Къде тук виждате командата **return**, с която една функция казва колко ще върне като резултат?

Вярно или невярно?

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( oc ) {
  if ((oc < 2) || (oc > 6))
     return “Не е оценка!“;
  else return oc;
}
```

Невярно!

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( oc ) {
  if ((oc < 2) || (oc > 6))
     return “Не е оценка!“;
  else return oc;
}
```

Къде тук виждате команда за въвеждане от клавиатурата?

Вярно!

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( ) {
  var oc = prompt( “Въведи
                оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6))
     return “Не е оценка!“;
  else return oc;
}
```

Тази функция прави точно каквото се иска в условието: **пита** за стойност и после с **return** я връща като резултат от функцията.

Ефективно!

Напишете функция, която позволява да въведете валидна оценка от клавиатурата и я връща към главната програма

```
function validnaOcenka( ) {
  var oc = prompt( “Въведи
                оценка от 2 до 6“);
  if ((oc < 2) || (oc > 6)) {
     alert(“Не е оценка!“);
     return validnaOcenka( ) ;
  }
  else return oc;
}
```

Тук дори даваме втори шанс да се въведе валидна оценка, ако първия път е въведено нещо грешно. Това извикване на функцията от самата себе си се нарича **рекурсия**.

Вярно или невярно?

Напишете функция за работа с точки, позволяваща извеждане на координатите на точката на екрана

```
function pechatTochka( ) {
  var x, y;
  x = prompt(“x = “);
  y = prompt(“y = “)
  alert(“(“ + x + “, “ + y + “)”);
  return x;
  return y;
}
```

Невярно!

Напишете функция за работа с точки, позволяваща извеждане на координатите на точката (подадени като параметри) на екрана

```
function pechatTochka( ) {
  var x, y;
  x = prompt("x = ");
  y = prompt("y = ");
  alert("(" + x + ", " + y + ")");
  return x;
  return y;
}
```

Тук ние **питаме** за **x** и **y**, а трябва да ги **подадем като параметри**.

В условието се иска да **изведем** координатите на екрана, а не да ги **върнем като резултат** с **return**.

Вярно!

Напишете функция за работа с точки, позволяваща извеждане на координатите на точката (подадени като параметър) на екрана (и не връща никакъв резултат)

```
function pechatTochka(x, y) {
  alert("(" + x + ", " + y + ")");
}
```

Вярно или невярно?

Напишете функция за работа с точки, позволяваща въвеждане на координатите на точката от клавиатурата

```
function vavediTochka( ) {
  var x, y;
  x = prompt(“x = “);
  y = prompt(“y = “)
  alert(“(“ + x + “, “ + y + “)”);
  return x;
  return y;
}
```

Невярно!

Напишете функция за работа с точки, позволяваща въвеждане на координатите на точката от клавиатурата

```
function vavediTochka( ) {
  var x, y;
  x = prompt("x = ");
  y = prompt("y = ");
  alert("(" + x + ", " + y + ")");
  return x;
  return y;
}
```

Това извеждане за какво е? Иска се **връщане** с **return**.

Една функция не може да върне едновременно две стойности по този начин. Още при първата команда **return** изпълнението на функцията приключва.

Вярно!

Напишете функция за работа с точки, позволяваща въвеждане на координатите на точката от клавиатурата (и ги връща като резултат)

```
function vavediTochka( ) {
  var tochka = { };
  tochka.x = prompt("x = ");
  tochka.y = prompt("y = ");
  return tochka;
}
```

Това, че създаваме обект…

…ни позволява да върнем чрез него повече от една стойности.

Вярно или невярно?

Напишете функция, която връща кое е по-голямото от две числа

```
function max2(a, b) {
  var max;
  if (a > b)
      max = 1;
  else max = 2;
  return max;
}
```

Невярно!

Напишете функция, която връща кое е по-голямото от две числа

А за a==b?

Има стойности на параметрите, за които не е ясно какво ще върне функцията.

```
function max2(a, b) {
  var max;
  if (a > b)
      max = 1;
  else max = 2;
  return max;
}
```

Вярно!

Напишете функция, която връща кое е по-голямото от две числа

А за a==b?

```
function max2(a, b) {
  var max=0;
  if (a > b)
      max=1;
  else if (a < b)
       max=2;
  return max;
}
```

Сега тук са обхванати и трите частни случая:
1. a > b
2. a < b
3. a == b

Проследете с примерни стойности на a и b, за да се убедите. че е така.

Вярно!

Напишете функция, която връща кое е по-голямото от две числа

А за a==b?

```
function max2(a, b) {
  if (a > b)
      return 1;
  else if (a < b)
      return 2;
  else return 0;
}
```

И тук са обхванати и трите частни случая:
1. a > b
2. a < b
3. a == b

В предния слайд първо се пресмята стойността, а после се връща. Тук директно се връща.

Вярно или невярно?

Напишете функция, която намира сумата на две числа

```
var a, b, sum;
function calcSum( ) {
  sum=a+b;
}
```

Неудачно!

Напишете функция, която намира сумата на две числа

```
var a, b, sum;
function calcSum( ) {
  sum=a+b;
}
a=5;
b=4;
calcSum( );
alert(sum);
```

Използването на глобални променливи за предаване на информация към функции е много неудобно.

Вярно или невярно?

Напишете функция, която намира сумата на две числа

```
function calcSum(a, b, sum) {
  sum=a+b;
}

var sum;
calcSum(4, 5, sum);
alert(sum);
```

Невярно!

Напишете функция, която намира сумата на две числа

```
function calcSum(a, b, sum) {
  sum=a+b;
}
```

Тук **sum** е подадена като **параметър**, така че резултатът в нея не се запомня и не се връща обратно.

```
var sum;
calcSum(4, 5, sum);
alert(sum);
```

Вярно!

Напишете функция, която намира сумата на две числа

```
function calcSum(a, b) {
  var sum;
  sum=a+b;
  return sum;
}

alert(calcSum(4, 5));
```

Колко по-кратко е записването, когато предаваме информацията чрез **параметри** и връщаме резултата с **return**!

Вярно!

Напишете функция, която намира сумата на две числа

```
function calcSum(a, b) {
  return a+b;
}

alert(calcSum(4, 5));
```

Може и така, става още по-кратко.

Вярно или невярно?

Напишете функции, които намират лицето и обиколка на кръг

```
var r;
const PI=3.14;
function calcLice( ) {
  return PI*r*r;
}
function calcObikolka( ) {
  return 2*PI*r;
}
```

Неудачно!

Напишете функции, които намират лицето и обиколка на кръг

Различните кръгове имат различен радиус. Не е удобно да ползваме глобална променлива за предаване на тази информация.

```
var r;
const PI=3.14;
function calcLice( ) {
  return PI*r*r;
}
function calcObikolka( ) {
  return 2*PI*r;
}
r = 10; alert(calcLice( ));
```

Вярно или невярно?

Напишете функции, които намират лицето и обиколка на кръг

```
function calcLice( r ) {
  const PI=3.14;
  return PI*r*r;
}
function calcObikolka( r ) {
  const PI=3.14;
  return 2*PI*r;
}
```

Неефективно!

Напишете функции, които намират лицето и обиколка на кръг

```
function calcLice( r ) {
  const PI=3.14;
  return PI*r*r;
}
function calcObikolka( r ) {
  const PI=3.14;
  return 2*PI*r;
}
```

Различните кръгове ползват **едно** и също число **PI**. Не е удачно да ползваме **локална** константа за тази информация.

Ефективно!

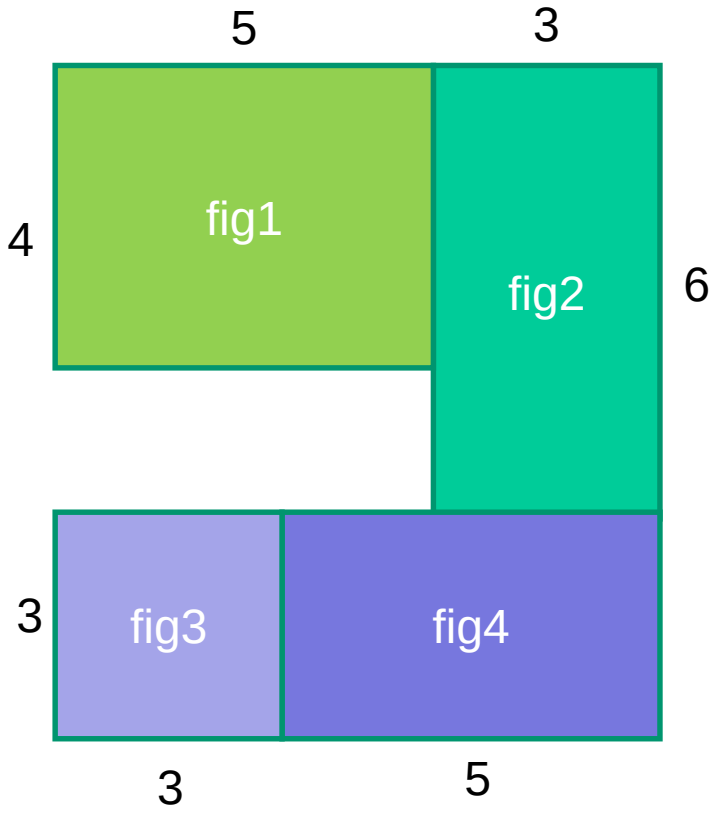
Напишете функции, които намират лицето и обиколка на кръг

Което е **различно**, го подаваме като **параметър** (виж **r**). Което е общо за всички функции, става глобална константа или променлива (виж PI).

```
const PI=3.14;
function calcLice( r ) {
  return PI*r*r;
}
function calcObikolka( r ) {
  return 2*PI*r;
}
alert(calcLice(10));
alert(calcObikolka(10));
```

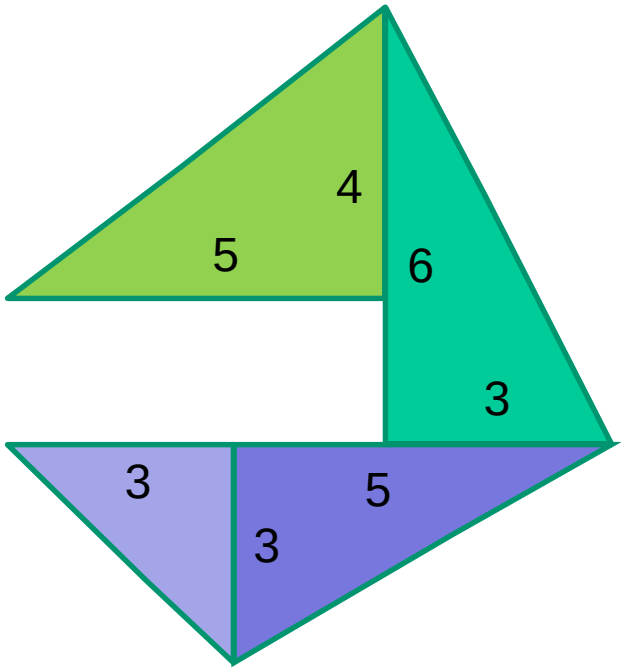
Пример за пресмятане на лицето на съставна фигура чрез функция

```
function lice(a, b)  {
   return a*b;
}
var fig1 = lice(5, 4);
alert( fig1+ lice(3, 6) +
        lice(3+5, 3) );
```



Можем да запомним резултата от извикването на една функция в променлива или директно да го ползваме в израз заедно с резултата от извикването на други функции. Може като параметър на функция да подаваме израз, който ще се пресметне, преди да се извика функцията.

Да се намери лицето на фигурата



1. Да се направи функция lice, която **връща** лицето на правоъгълен триъгълник по дадени катети **a** и **b**
2. Да се направи друга функция izvedi, която **извежда** съобщение: "Площта на фигурата е **X** кв.см.", където **X** е подадено като параметър.
3. С помощта на двете функции да се **изведе** колко е площта на голямата фигура.
4. Да се **изведе** площта на всеки от малките триъгълници с помощта на функцията izvedi.
5. Да се направи функция infoZaTriagalnik, която **извежда** катетите на триъгълник и лицето му.

```
function име(параметри) {
    команди;
    return резултат;
}

име(параметри);
```

Задачи

Напишете програма, демонстрираща следните функции:

1. Функция за проверка дали една правоъгълник е квадрат
2. Функции за пресмятане на лице и обиколка на правоъгълник
3. Функция, която получава като параметри страните на правоъгълник и извежда дали той е квадрат и колко е лицето и обиколката му

```
function име(параметри) {
    команди;
    return резултат;
}

име(параметри);
```

Задачи

4. Функция за проверка дали 3 числа са валидни страни на триъгълник
5. * Функция за проверка вида на триъгълник – равнобедрен, равностранен, правоъгълен, друг
6. Функции за пресмятане на лице и обиколка на триъгълник по подадени 3 страни
7. Функция, която получава три числа и извежда пълната информация: дали са страни на триъгълник, видът и лицето и обиколката му

Край