



#ШПАРГАЛОЧКИ

# СОЗДАНИЕ САЙТОВ FRONT-END РАЗРАБОТКА

## Средний уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити  
в Telegram: [https://t.me/hw\\_school](https://t.me/hw_school)





# Игра - "Угадай число"





# Условия

Когда код нужно выполнять не всегда, а только при выполнении какого-то условия (например, если пользователь победил - выводить сообщение об этом), нужна специальная команда - **оператор if**.

**if** - это условный оператор, который проверяет, **истинно ли утверждение**, идущее в круглых скобках после него. Код, который должен сработать при выполнении условия, заключается в фигурные скобки:

```
if (secretNumber == number) {  
  
    alert("Победа!");  
  
}
```



# Условия

В скобках после **if** можно проверять не только равенство, но и больше/меньше ли одно число другого. Используемые операторы:

**==** - для проверки равенства нужно 2 знака равно

**<** - меньше

**>** - больше

При сравнении 2 значений ответ всегда будет **true** (если сравнение истинно) или **false** (если сравнение ложно). Проверить это можно с помощью `console.log()`:

```
console.log(3 == 3); // true
```

```
console.log(5 > 7); // false
```



# Функции

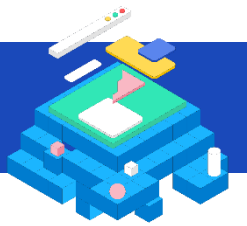
**Функция** - это блок кода, который можно вызвать (заставить работать) из любого места программы. Функции могут существовать без имени (как в случае с **onclick**), но если предполагается использовать функцию в другом месте программы, имя ей все-таки нужно.

Чтобы **создать** функцию, нужно ключевое слово **function**, после которого указывается имя функции, круглые скобки и код внутри фигурных скобок:

```
function sayHello() {  
  
    console.log("Hello!");  
  
}
```

А чтобы функция заработала, нужно ее **вызвать** (написать имя функции и круглые скобки):

```
sayHello();
```



# Math

**Math** - специальный объект, в котором хранятся переменные и функции, связанные с математикой. Чтобы “достать” их из Math, нужно написать их через точку:

**Math.PI**; // число Пи

А также в Math хранятся интересные функции, например:

**Math.random()** - создает случайное число от 0 до 1

**Math.floor()** - округляет число (в меньшую сторону)



# Принимающая функция и return

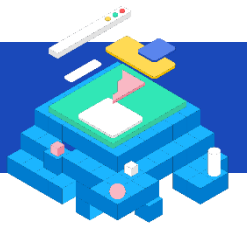
Чтобы функция могла работать с какими-то значениями из основной программы, их нужно **передавать** в нее. А значит, при создании функции нужно указать в ее круглых скобках, что она может **принимать** (тут указываются переменные).

Кроме того, функции могут **возвращать** что-то в место своего вызова с помощью команды **return**:

```
function getRandom(min, max) {  
  
    return Math.random() * (max - min + 1) + min; // случайное число от min до max  
  
}
```

При вызове функции ей уже нужно **передать** значения (например, числа). А чтобы сохранить то, что функция **вернет**, перед ее вызовом нужно поставить переменную и знак "=":

```
let randomNumber = getRandom(1, 10); // в randomNumber сохранится случайное число от 1 до 10
```



# disabled и focus

С помощью JavaScript можно не только получать и изменять содержимое html-элементов, но и менять их состояние. Например:

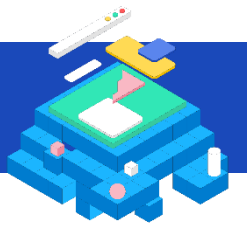
**disabled** - делает элемент (например, кнопку) неактивным

```
document.querySelector("button").disabled = true;
```

**focus()** - работает с текстовыми полями, автоматически ставит фокус (курсор) на это поле:

```
document.querySelector(".number1").focus();
```





# innerHTML

**innerHTML** - свойство, позволяющее вставить в html-элемент какой-то текст:

```
document.querySelector(".answer").innerHTML = "Вы угадали";
```

Важно учитывать, что **innerHTML** не добавляет текст к уже существующему, а именно **заменяет** содержимое html-элемента на новое.