Функция – блок кода, который определяется один раз и может выполняться многократно.

При определении функции можно задать параметры. ()

При вызове функции на месте параметров указываются параметры.

В JS любая функция:

* получает при вызове доп. аргумент – контекст вызова
* всегда возвращает некое значение
* сама является объектом
* может выступать в качестве подпрограммы, метода объекта, конструктора объекта

Первый способ создания функции – инструкция function.

Инструкции function могут быть вложены в другие определения функций (должен быть первый уровень вложенности – не циклы, не условные инструкции).

Если в JS объявлять функции с одним именем, последнее объявление перекрывает все предыдущие.

В ES2015 появилось выражение стрелочных функций – короткий аналог выражений function.

Стрелочные функции не имеют своего this. Внутри функций-стрелок – тот же this, что и снаружи.

Стрелочные функции нельзя использовать в качестве конструктора, то есть нельзя вызывать через new.

Стрелочные функции не имеют своего аргумента. В качестве аргумента используются аргументы внешней «обычной функции».

В JS распространен следующий прием: нужные аргументы передаются функции через свойства объекта.

Объект, созданный конструктором Arguments, используется для хранения всех аргументов, переданных функции при вызове. Этот объект подобен массиву. В теле любой функции этот объект доступен через идентификатор arguments (обращение вне функции генерирует ошибку).

Функция может содержать инструкцию return для выхода и возврата некоего значения.

В JS функции могут вызываться четырьмя способами:

* Как функции
* Как методы
* Как конструкторы
* Косвенно, с помощью методов call.

Метод – функция, которая хранится в свойстве объекта.

Вызов метода = выражение обращения к свойству + выражение вызова.

Контекст вызова определяется именно способом вызова функции (а не местом вызова). Это важно, если функция, которая вызывается как метод, вызывает свою вложенную функцию как метод.

Вызов функции как конструктора – это вызов функции, который предварен ключевым словом new:

1. Создается новый объект и назначается в качестве this
2. Отрабатывает функция
3. Возвращаемым значением функции всегда будет объект – либо объект после return, либо созданный на шаге 1.

Если функция вызывается как конструктор и не имеет аргументов, можно не указывать круглые скобки.

У new без скобок более низкий приоритет, чем у new со скобками.

Любая функция – это особый объект. У этого объекта есть методы call() и apply().

Вызов функции метода bind() возвращает новую функцию, у которой контекст вызова this жестко установлен в указанное значение.

**Замыкание**

Если функция F() возвращает свою вложенную функцию g(), то все переменные из scope F() (и их значения) будут доступны в g(). Это явление называется замыканием (closure).

В современном JS распространен такой прием: объявляется выражение-функция (без имени), и сразу же происходит вызов этой функции.