Lesson - 15

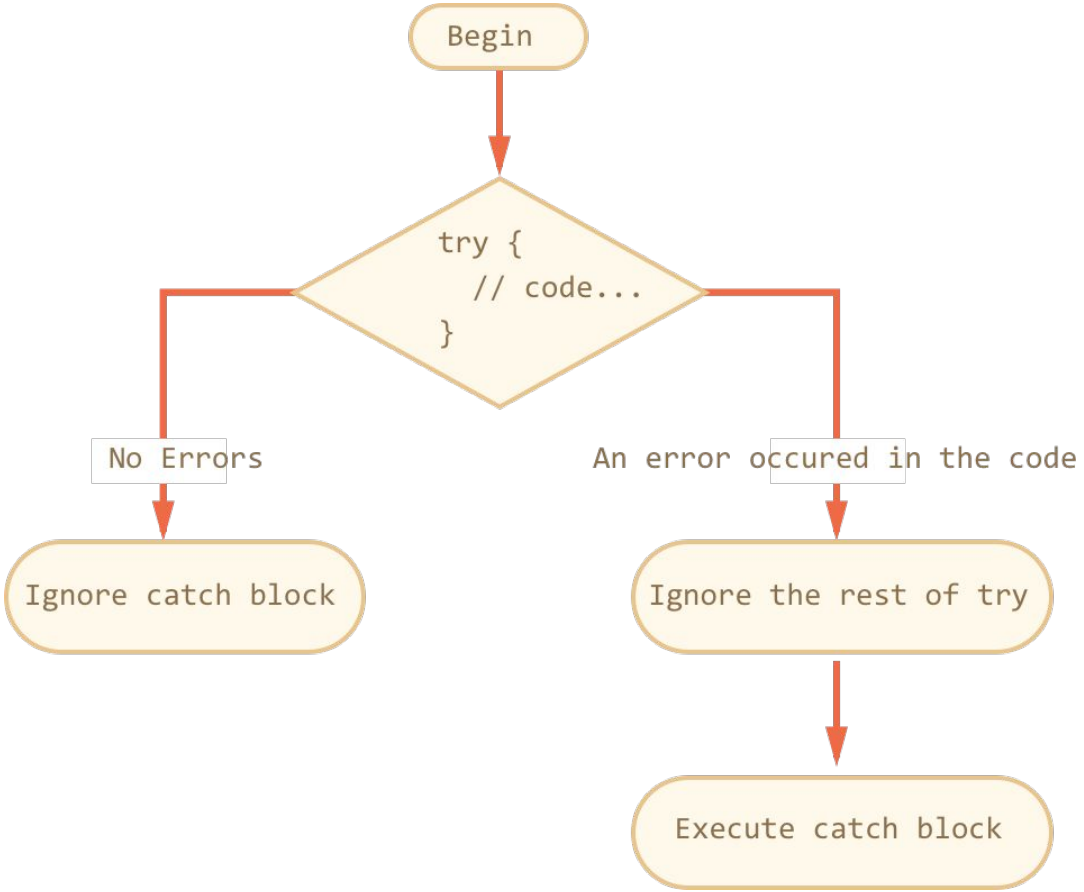
Scope. Hoisting. Functions.

SkillUp, by Vitali Cernomschi

План занятия

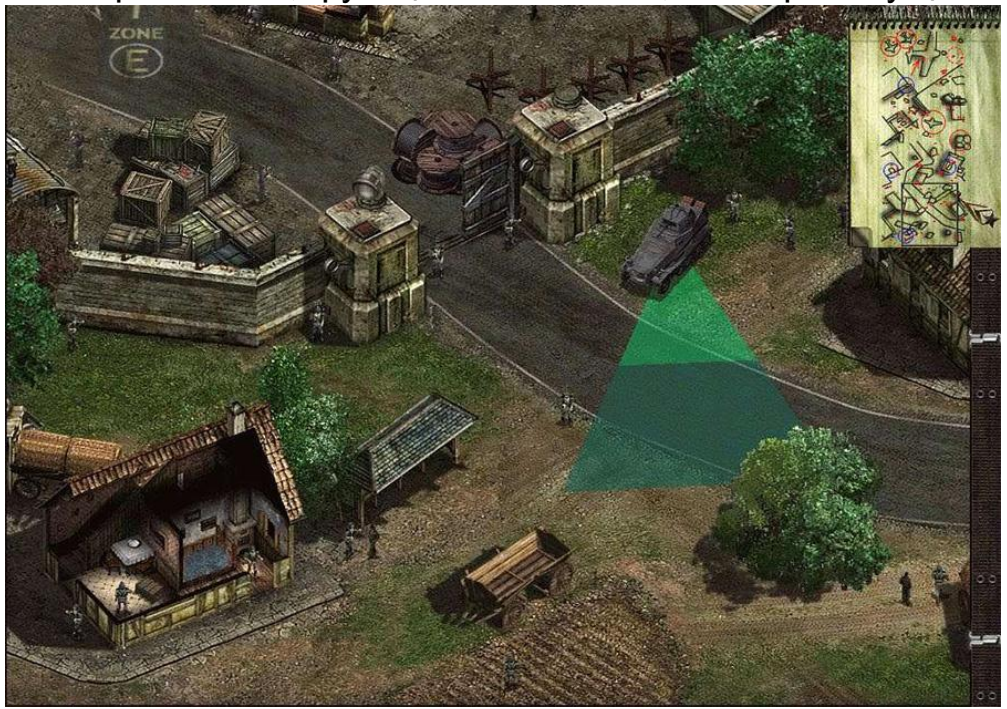
1. Мои вопросы по домашнему заданию.
2. Try - catch - finally
3. Scope and hoisting
4. Functions (FD, FE, NFE)

Try Catch


Begin
try {
// code...
}
No Errors
An error occured in the code
Ignore catch block
Ignore the rest of try
Execute catch block

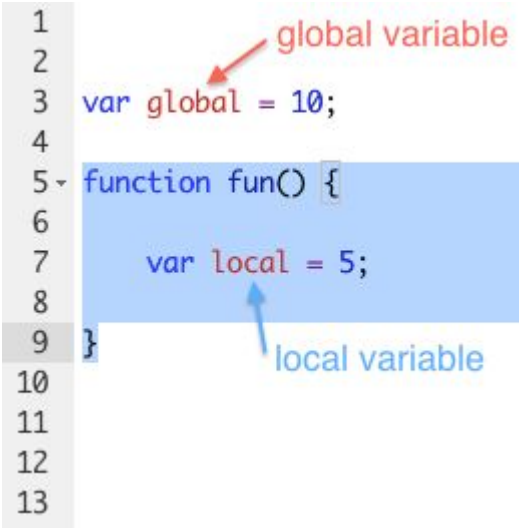
Scope

Область видимости - это совокупность переменных, функций и объектов, к которым существует доступ во время его исполнения.



Scopes

- Global scope
- Function scope (var)
- Block scope (let, const - are not hoisted, will be clarified as part of ES6)
- Eval scope



Global Scope

Глобальная область видимости (ГОВ) - ваш лучший друг и худший кошмар. Учась работе с различными областями видимостями, вы не встретите проблем с ГОВ, разве что вы увидите пересечения имен.

Часто можно услышать «глобальная ОВ - это плохо», но нечасто можно получить объяснение - почему.

ГОВ - не плохо, вам нужно ее использовать при создании модулей и API, которые будут доступны с разных ОВ, просто нужно использовать ее на пользу и аккуратно.

Global Scope

Вам предстоит использовать различные библиотеки: jQuery, lodash/underscore и тд:

```
$( '.myClass');
```

Мы получаем доступ к jQuery в глобальной области видимости, также называемую пространством имен.

Global object

Глобальный объект представляет собой обычный объект, который создается автоматически при запуске интерпретатора.

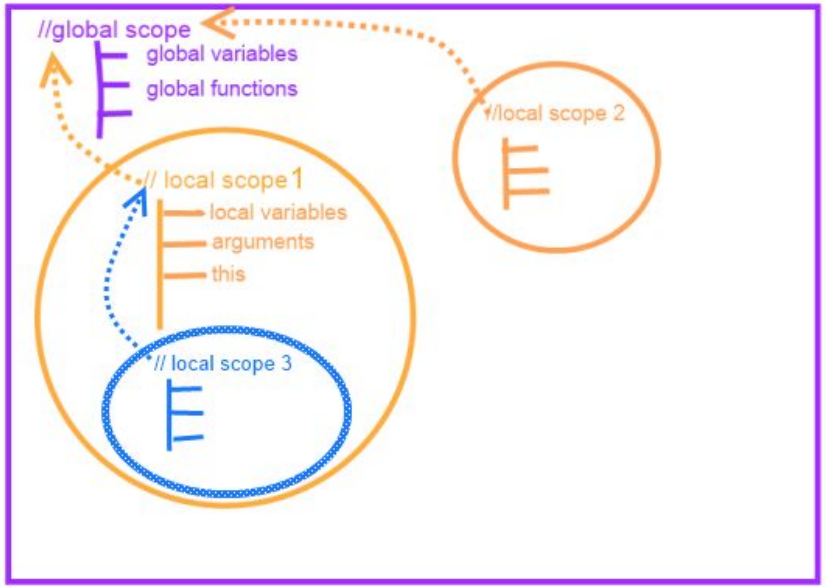
В JavaScript роль глобального объекта играет объект Window.

Этот объект имеет свойство window, ссылающегося на сам объект Window.

Объект Window одновременно является и глобальным объектом и кроме этого содержит ряд собственных свойств и методов для работы с окном браузера.

Local Scope

Локальной ОВ называют любую ОВ, определенную после глобальной. Обычно у нас есть одна ГОВ, и каждая функция несет в себе локальную ОВ. Каждая функция, определенная внутри другой функции, имеет свою локальную ОВ, связанную с ОВ внешней функции.



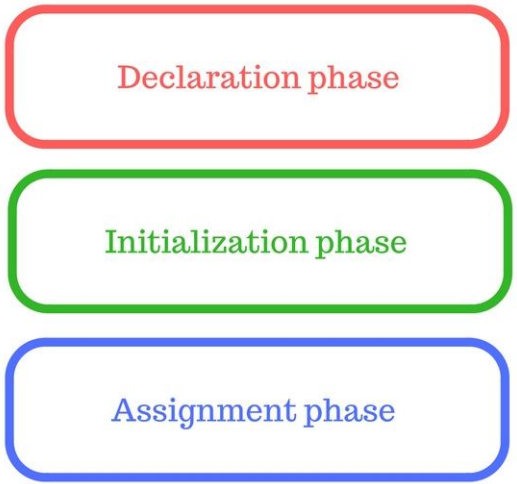
Local Scope

Все переменные с ЛОВ не видны в ГОВ. К ним нельзя получить доступ извне непосредственно:

```
var myFunction = function () {
var name = 'Todd';
console.log(name); // Todd
};
// ReferenceError: name is not defined
console.log(name);
```

Scope and Hoisting

Variables lifecycle




```
// var hoisting
num;      // => undefined
var num;
num = 10;
num;      // => 10
// function hoisting
getPi;    // => function getPi() {...}
getPi(); // => 3.14
function getPi() {
  return 3.14;
}
```

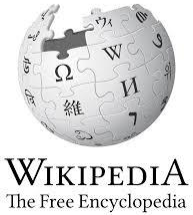
Function definition

A JavaScript function is a block of code designed to perform a particular task. A JavaScript function is executed when "something" invokes it (calls it).

Reference: https://www.w3schools.com/js/js_functions.asp

Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure—a set of statements that performs a task or calculates a value. To use a function, you must define it somewhere in the scope from which you wish to call it.

Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions

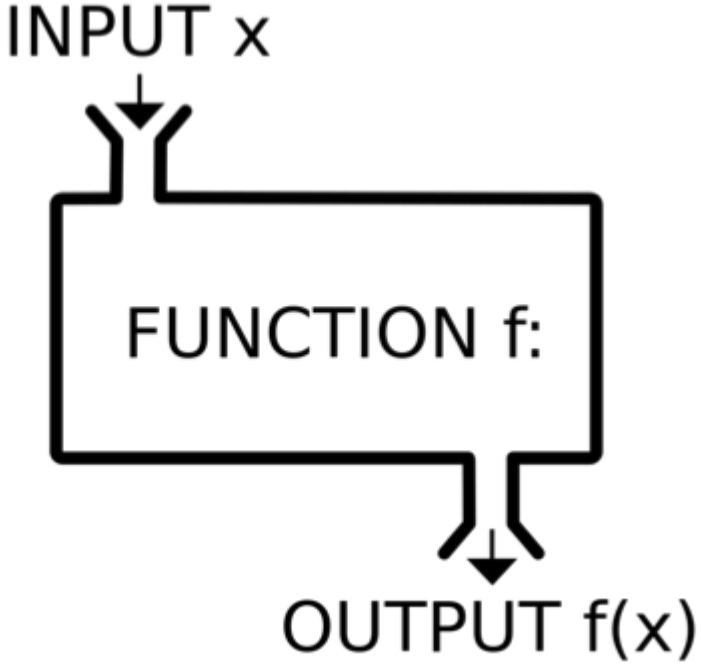


Фу́нкция в программировании — фрагмент программного кода (подпрограмма), к которому можно обратиться из другого места программы.

Reference: wiki

Function

Functions are considered as **First Class citizen in JavaScript**



Ways to create functions

1. Function Declaration

```
function [function name](param1, param2, ...param3) {
    // Function Body & Logic
}
```

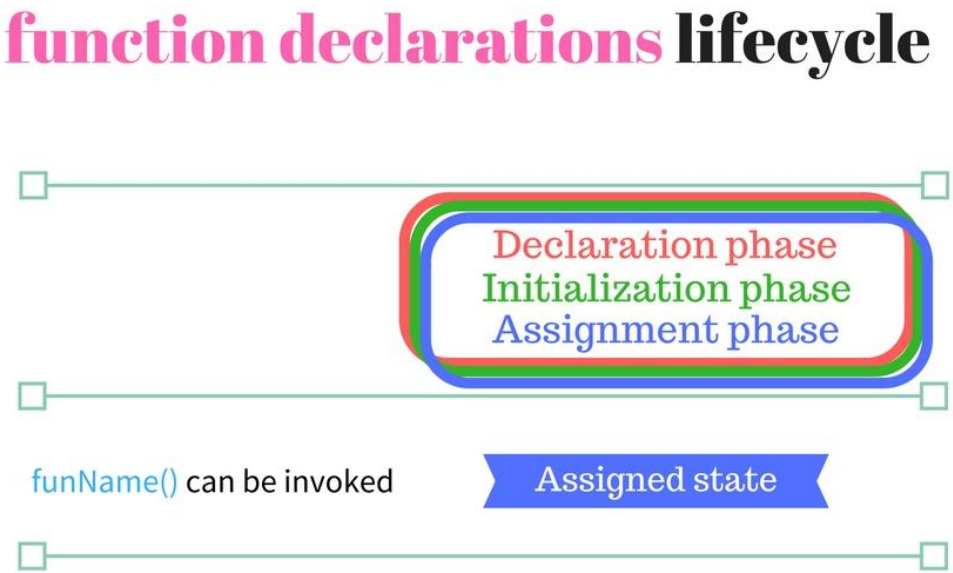
2. Function Expression

```
var [variable name] = function(param1, param2, ...param3) {
    // Function Body & Logic
}
```

3. Named Function Expression

```
var [variable name] = function [function name](param1, param2) {
    return param1 + param2 ;
}
```

function declaration lifecycle



References

1. Try - catch
   RU: https://learn.javascript.ru/exception

2. Functions
   RU: https://learn.javascript.ru/function-basics
   https://learn.javascript.ru/function-declaration-expression
   https://learn.javascript.ru/named-function-expression

   EN: https://javascript.info/

3. FE vs FD:
https://medium.com/@raviroshan.talk/javascript-function-declaration-vs-expression-f5873b8c7b38

Домашнее задание (* - advanced)

1. Нужно прочитать подразделы 2.17-2.21 RU версии или 2.14-2.16 в EN версии

   RU version: https://learn.javascript.ru/first-steps

   EN version: https://javascript.info/first-steps

2. Написать функцию на JS вычисляющая дискриминант.

3*. Написать функцию на JS вычисляющая корни квадратного уравнения.