

Условни конструкции в JavaScript

Димитър Митев

Условни конструкции

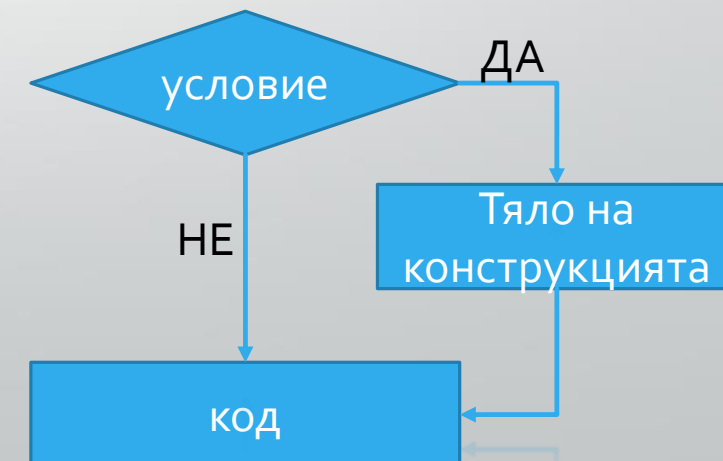
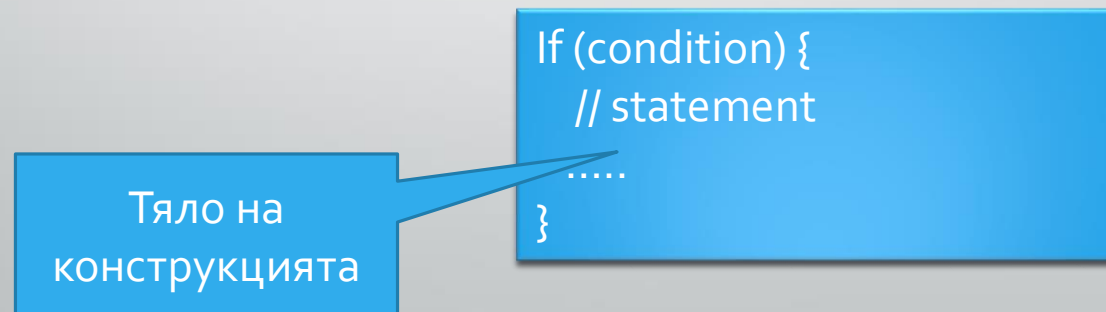
- Условните конструкции дават възможност да се управлява логиката и начина на изпълнение на кода
- Условните конструкции оценяват дадено твърдение като булева стойност и на база на резултата извикават/изпълняват различни парчета от кода

Условие и тяло

- Условните конструкции в JS могат да имат условие от какъвто и да е тип
- Условиата винаги се оценяват до булев тип (*true-like* или *false-like*)
- Тялото на условната конструкция може да бъде всякакъв израз завърващ с ; или съвкупност от от изрази, завърващи с ;
- Добра практика е тялото на конструкцията да бъде заградено в {}

Условна конструкция if

- Най-простата условна конструкция
- Проверява дали е изпълнено дадено условие
 - Ако е изпълнено се изпълнява кода, който се намира в тялото на if конструкцията, преди да продължи изпълнението на кода
 - Ако не е изпълнено условието, тялото на if конструкцията се пропуска и кода продължава със своето изпълнение



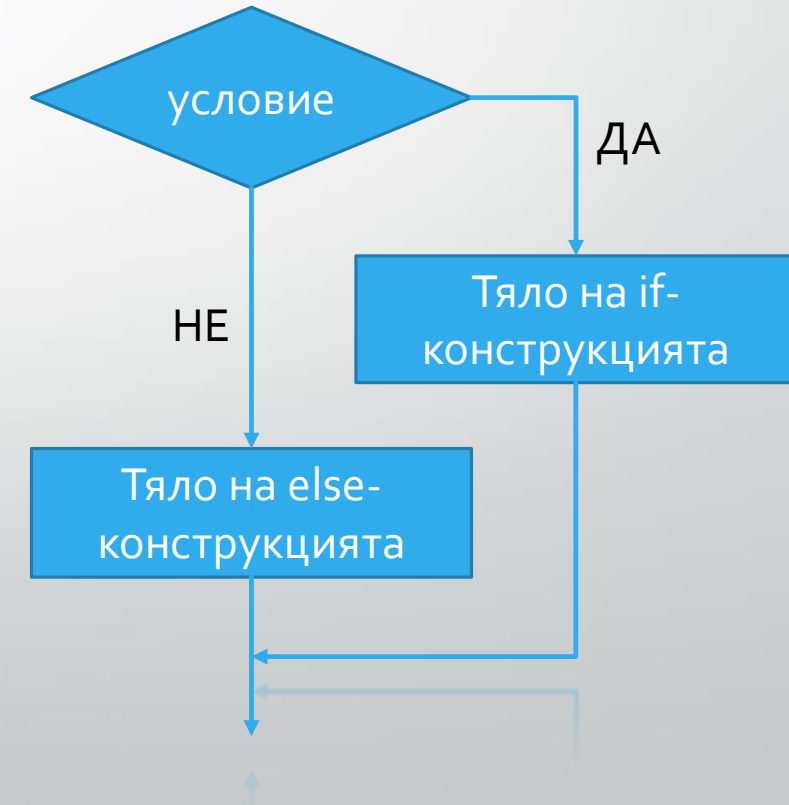
if конструкция. Пример

```
var sideA = 5;  
var sideB = 7;  
var biggerSide;  
  
if (sideB > sideA) {  
    biggerSide = sideB;  
}
```

if-else конструкция

- 1 идея по-сложна от *if*-конструкцията
- Оценява се условите и ако то е вярно са изпълнява, тялото на *if*-конструкцията, в противен случай се изпълнява тялото на *else*-конструкцията

```
if (expression) {  
    statement1;  
} else {  
    statement2;  
}
```



if-else конструкция. Пример

```
var number = 27;  
  
if (number % 2) {  
    console.log('Odd number');  
} else {  
    console.log('Even number');  
}
```

Вложени *if-else* конструкции

- Във всяка *if* или *else*-конструкция може да бъде вложена друга *if* или *if-else* конструкция
- Всеки *else* принадлежи към най-близкия, предхождащ *if*

```
If (expression1) {  
    if (expression2) {  
        statement1;  
    } else {  
        statement2;  
    }  
} else {  
    statement3;  
}
```


Множество *if-else* конструкции

- Често се налага проверка на повече от 1 условие
- Може да се използва съкратено записване *} else if { ... }* което е еквивалентно на влягането на *if*-конструкция в *else*-конструкцията

```
If (expression1) {  
    statement1;  
} else if (expression2) {  
    statement2;  
} else {  
    statement3;  
}
```

switch-case конструкция

- Позволява ни да правим редица сравнения наведнъж
- Съдържа списък от действия и конкретни условия, при които да се изпълнят
- Оценява се отгоре надолу, т.е. кодът, който се намира в 1-вото вярно твърдение, ще бъде изпълнен пръв
- Може да има дефолтно поведение
- *Case*-овете трябва да бъдат константи стойности
- Всеки *case* трябва да бъде прекъснат (*break* или *return*), в противен случай кода продължава изпълнението си надолу



switch-case конструкция(1)

- В JS се поддържа т.нар. fall-through поведение, т.е. ако дадено условие е пропуснато се оценява до най-близкия *break/return*
- *switch/case* конструкцията не се счита за добър подход и се препоръчва да не се използва
- Води до усложнено четене на кода

```
switch (day) {  
  case 1: console.log('Monday'); break;  
  case 2: console.log('Tuesday'); break;  
  case 3: console.log('Wednesday'); break;  
  case 4: console.log('Thursday'); break;  
  case 5: console.log('Friday'); break;  
  case 6: console.log('Saturday'); break;  
  case 7: console.log('Sunday'); break;  
  default: console.log('Error!'); break;  
}
```

true-like & *false-like* стойности

- Всеки тип в JS може да бъде оценяван като булева променлива
- Някои стойности винаги се считат за *false*: *false*, 0 (нула), "", *null*, *undefined*, *NaN*
- Всички останали се оценяват до *true*
- [Прочетете повече](#)

НЯКОЙ ИМА ЛИ



ВЪПРОСИ?

Домашна работа

1. Напишете JS код, който по зададена цифра, изписва нейното наименование. Например: 1 -> one
2. Напишете JS код, който по заведени 3 числа намира най-малкото
3. Напишете JS код, който сортира 3 числа във възходящ ред
4. Напишете JS код, който да намира корените на квадратно уравнение $ax^2 + bx + c = 0$ по зададени a , b и c . *Заб.: Квадратното уравнение може да има 0,1 или 2 реални корена
5. *Напишете програма, която конвертира дадено 3-цифрено число към неговата текстова репрезентация. Напр.: 312 -> three hundred and twelve