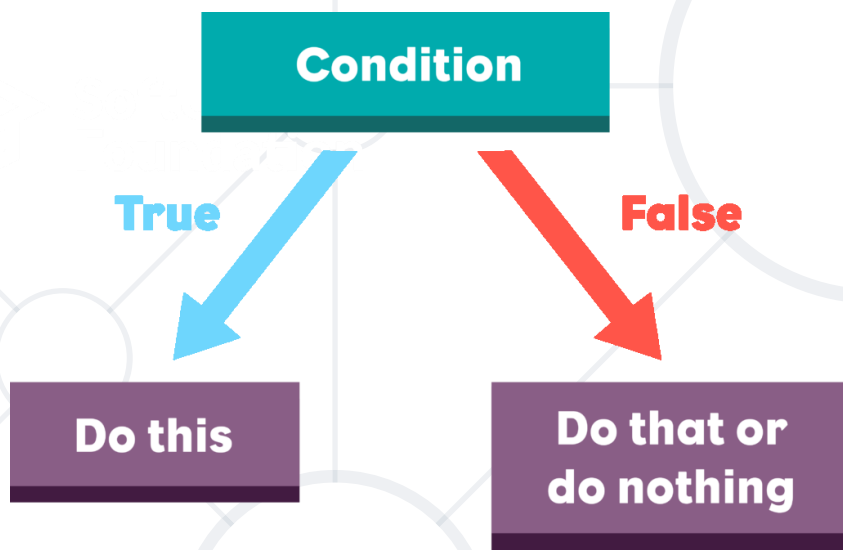


Проверки

Логически изрази и проверки. Условна конструкция if-else



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Software University

<http://softuni.bg>

Имате въпроси?

sli.do

#pb-jan

1. Преговор
2. Логически изрази и проверки
 - Оператори за сравнение.
3. Условни конструкции
4. Серия от проверки
5. Живот на променлива
6. Дебъгване
7. Решаване на изпитна задача





Преговор

1. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log('a' + 'b');
```

ab

Error

ba

195

1. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log('a' + 'b');
```

ab

Error

ba

195

2. Какъв е типът на променливата:

```
let number = "1000";
```

string

int

char

double

2. Какъв е типът на променливата:

```
let number = "1000";
```

string

int

char

double

3. Как се нарича долепването на два текста (низа)?



Събиране

Конкатенация

Кулминация

Съединяване

3. Как се нарича долепването на два текста (низа)?



Събиране

Конкатенация

Кулминация

Съединяване

4. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log(10 % 3);
```

10

1

0

3

4. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
console.log(10 % 3);
```

10

1

0

3

5. Каква стойност държи променливата **result**:

```
let a = 5;  
let b = 2;  
let result = a / b;
```

2

7

2.5

1

5. Каква стойност държи променливата **result**:

```
let a = 5;  
let b = 2;  
let result = a / b;
```

2

7

2.5

1

6. Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

```
console.log(1 + 1 + "4" + 2 + 1);
```

Error

243

9

2421

6. Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

```
console.log(1 + 1 + "4" + 2 + 1);
```

Error

243

9

2421



Логически изрази и проверки

Оператори за сравнение

Оператори за сравнение



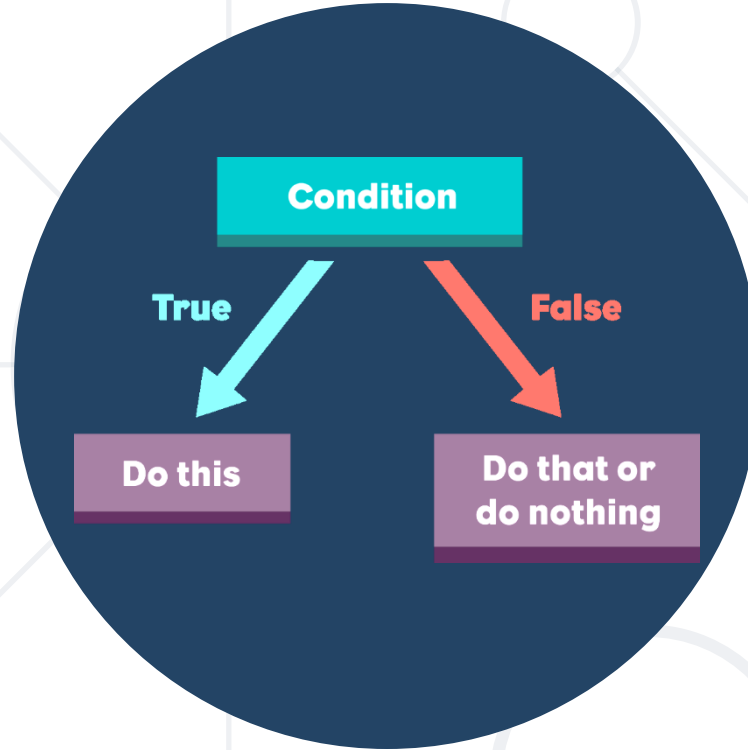
Оператор	Означение
Равенство по стойност (и тип данни)	==, ===
Различно по стойност (и тип данни)	!=, !==
По-голямо	>
По-голямо или равно	>=
По-малко	<
По-малко или равно	<=



- В програмирането можем да сравняваме стойности
 - Резултатът от логическите изрази е **true** или **false**

```
let a = 5;  
let b = 10;  
console.log(a < b);           // true  
console.log(a > 0);           // true  
console.log(a > 100);         // false  
console.log(a < a);           // false  
console.log(a <= 5);          // true  
console.log(b == 2 * a);      // true  
console.log("2" === 2);       // false
```






Условни конструкции

Прости проверки

Прости проверки

- Често проверяваме условия и извършваме действия според резултата



Условие
(булев израз)

```
if (...) {  
    // код за  
}
```

Код за изпълнение при
извършване на условието

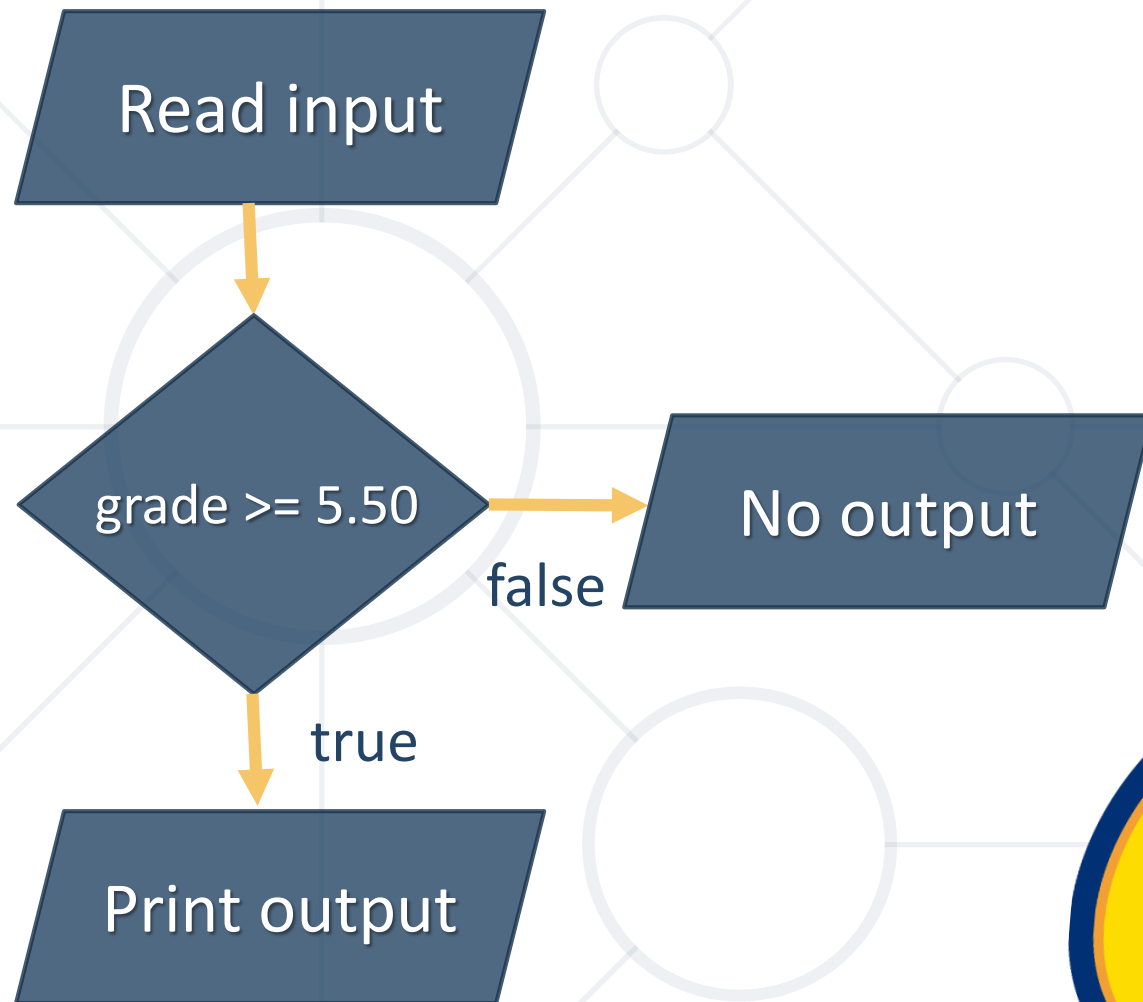
- Резултатът е **true** или **false**

- Напишете **програма**, която:
 - **Получава** оценка (**число**), въведена от потребителя
 - **Проверява** дали е отлична
 - **Отпечатва на конзолата** "Excellent!", ако оценката е по-голяма или равна на 5.50

- Пример:



236



6



Прости проверки – if-else

- При **невярност** (false) на условието, можем да изпълним други действия – чрез **else** конструкция

```
if (...) {  
    // код за изпълнение  
} else {  
    // код за изпълнение  
}
```

Код за изпълнение
при невярност на
условието



- Къдравите скоби { } въвеждат блок от код (група команди)
 - Ако конструкция **if** няма скоби, се изпълнява само следващият ред

```
let color = "red";  
if (color === "red")  
    console.log("tomato");  
else  
    console.log("banana");  
console.log("bye");
```

Изпълнява се винаги – не е част от
if/else конструкцията

```
let color = "red";  
if (color === "red") {  
    console.log("tomato");  
} else {  
    console.log("banana");  
    console.log("bye");  
}
```

По-голямото число – условие

- Напишете програма, която:
 - Получава две **числа**
 - Извежда **"Greater number: "**
 - Отпечатва на конзолата **по-голямото** от тях
- Пример:

5
8



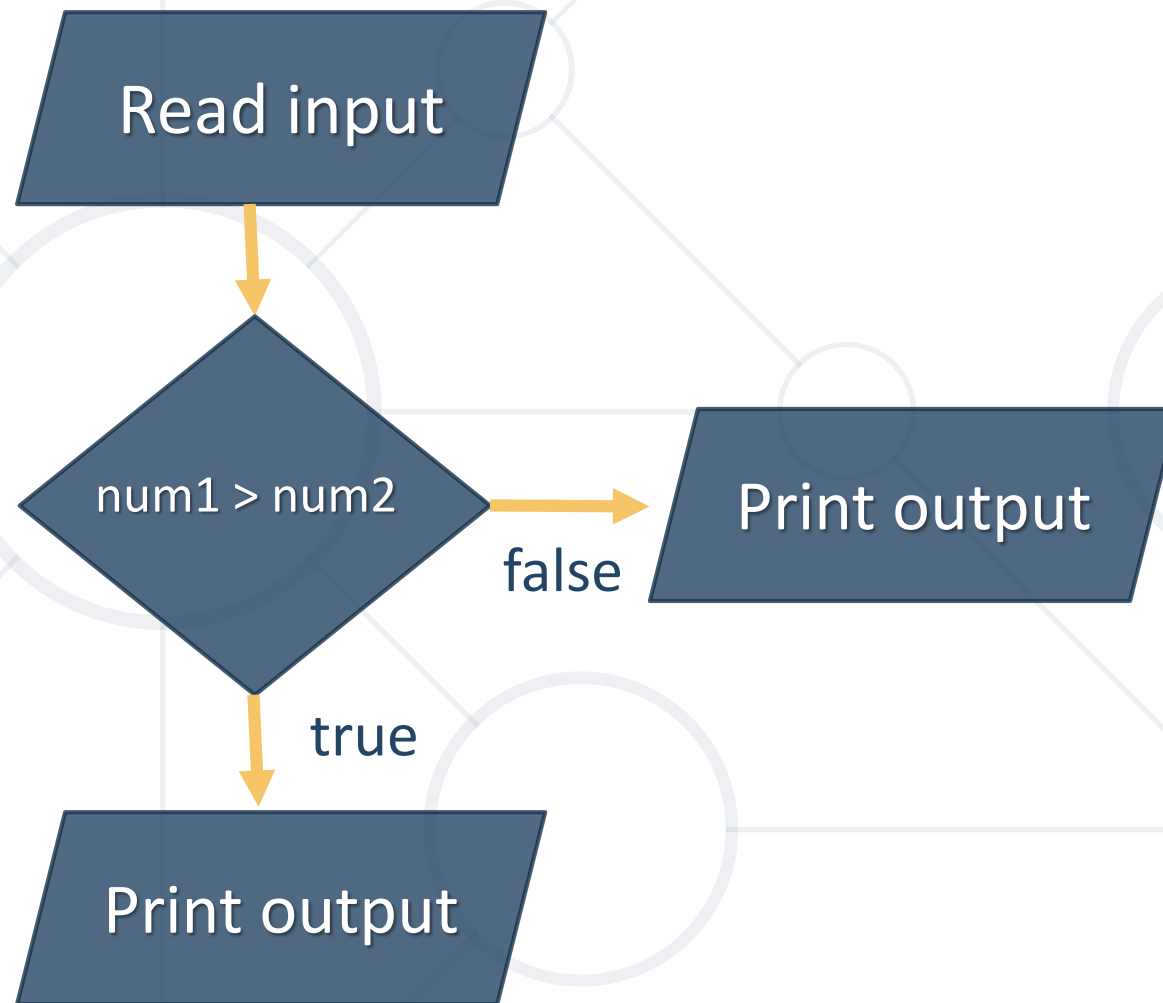
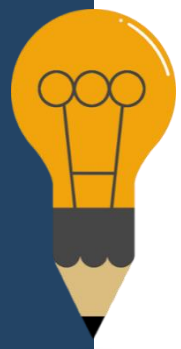
8

7
3



7

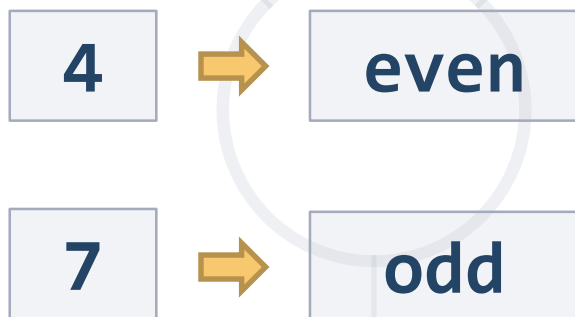




Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1012#2>

Четно или нечетно число – условие

- Напишете програма, която:
 - Проверява, дали едно число е **четно** или **нечетно**
 - Ако е четно, отпечатва на конзолата **"even"**
 - Ако е нечетно, отпечатва на конзолата **"odd"**
- Пример:



4 **7**

Четно или нечетно – решение

```
function isEven(input) {  
  let num = parseInt(input.shift());  
  if (num % 2 == 0) {  
    console.log("even");  
  } else {  
    console.log("odd");  
  }  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Compete/Index/1012#2>




Серии от проверки

По-сложни условни конструкции

Сerii от проверки

- Конструкцията **if/else-if/else...** е серия от проверки




```
if (...)  
    // код за изпълнение  
else if (...)  
    // код за изпълнение  
else if (...)  
    // код
```

TRUE OR **FALSE?**

- При истинност на едно условие, **не се продължава** към проверяване на следващите условия

Серия от проверки - пример

- Програмата проверява първото условие, установява, че е вярно и приключва



```
let a = 7;  
if (a > 4)  
    console.log("Bigger than 4");  
else if (a > 5)  
    console.log("Bigger than 5");  
else  
    console.log ("Equal to 7");
```

Извежда на
конзолата само
"Bigger than 4"

Число от 1 до 9 с текст - условие

- Напишете програма, която:
 - Чете **цяло число**, въведено от потребителя
 - Проверява неговата стойност [1,9]
 - Ако числото е по-голямо от 9 отпечатва "number too big"
 - Отпечатва стойността **с текст**
- Пример:

7 → seven

10 → number too big

Число от 1 до 9 с текст - решение

```
function number0to9([arg1]) {  
  let num = parseInt(arg1);  
  if (num === 1)  
    console.log("one");  
  else if (num === 2)  
    console.log("two");  
  // TODO: add more checks  
  else  
    console.log("number too big");  
}
```

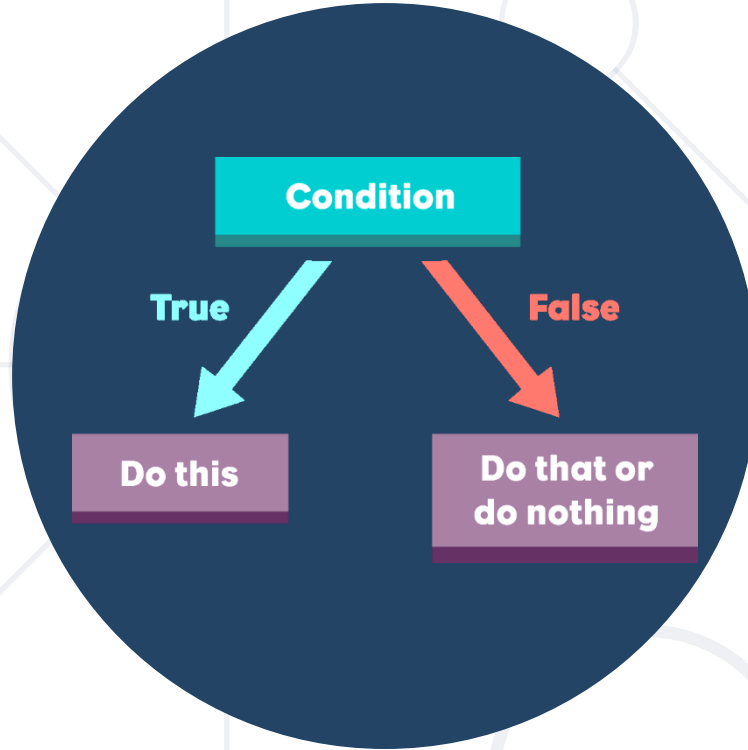


Живот на променлива

Диапазон на използване

- Обхват, в който може да бъде използвана
 - Пример: Променливата **salary** съществува **само** в блока от код на **if**-конструкцията

```
let currentDay = "Monday";  
if (currentDay === "Monday") {  
    let salary = Number(input.shift());  
}  
console.log(salary); // Error!
```



УСЛОВНИ КОНСТРУКЦИИ

Решаване на задачи в клас (лаб)

- Напишете програма, която:
 - Получава **вид** на **геометрична фигура** ("square", "rectangle", "circle" или "triangle")
 - Пресмята **лицето** спрямо вида на фигурата
- Примерен вход и изход:

square
5



25

rectangle
7
2.5



17.5

```
let shape = input.shift();
let area = 0;
if(shape === "square")
    let side = Number(input.shift());
    area = side * side;
else if(shape === "rectangle")
    let sideA = Number(input.shift());
    let sideB = Number(input.shift());
    area = sideA * sideB;
//TODO: add more conditions
console.log(area);
```



Дебъгване

Прости операции с дебъгер

Дебъгване

- Процес на проследяване на изпълнението на програмата
 - Това ни позволява да откриваме грешки (бъгове)



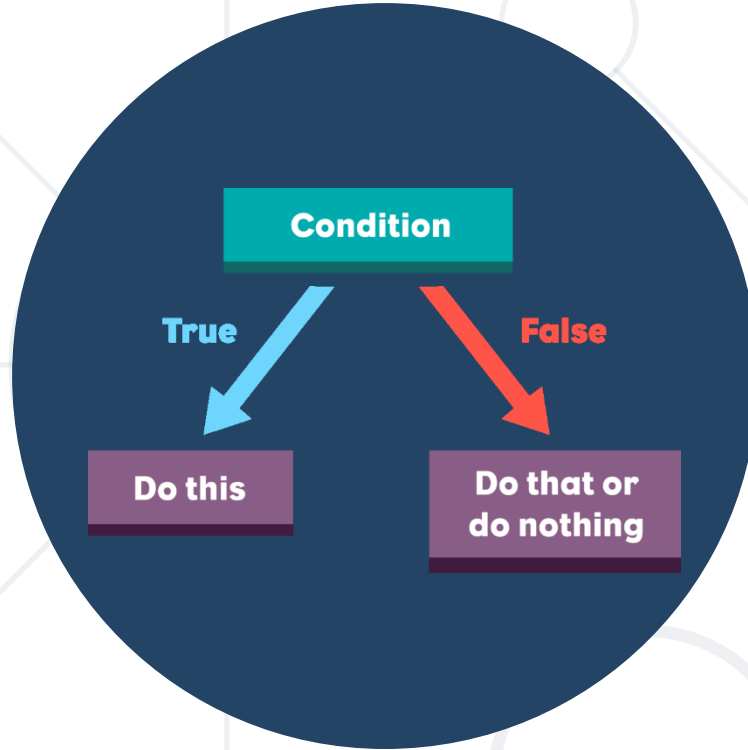
Breakpoint

```
1 function solve() {  
2   let currentDay = "Monday";  
3   let salary = 0;  
4   if (currentDay === "Monday") {  
5     salary = 100;  
6   }  
7   console.log(salary);  
8 }
```

Дебъгване във Visual Studio Code

- Натискане на [F5] ще стартира програмата в debug режим
- Можем да преминем към следващата стъпка с [F10]
- Можем да създаваме [F9] стопери – breakpoints
 - До тях можем директно да стигнем използвайки [Shift + F11]





УСЛОВНИ КОНСТРУКЦИИ

Решаване на задачи в клас (лаб)

- Конструкции за проверка на условие – **if** и **if-else**
- Живот на променливата
- Дебъгване



Въпроси?



SoftUni



**Software
University**



**SoftUni
Svetlina**



**SoftUni
Creative**



**SoftUni
Digital**



**SoftUni
Foundation**



**SoftUni
Kids**

SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре



INDEAVR

Serving the high achievers



INFRAGISTICS®



STEMO®
Computer Systems & Software

SUPERHOSTING.BG

SoftUni Organizational Partners

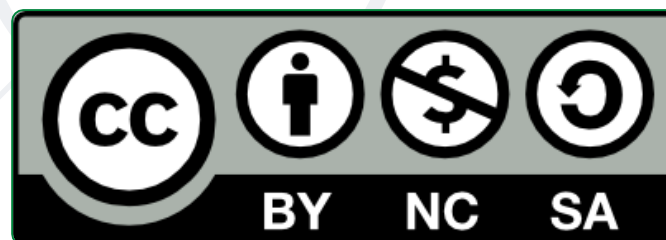


OneE
SOFTWARE



WORLD
OF
MYTHS

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
 - Книга "Основи на програмирането с JavaScript" от Светлин Наков и колектив с лиценз CC-BY-NC-SA

Обучения в СофтУни

- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



**Software
University**

