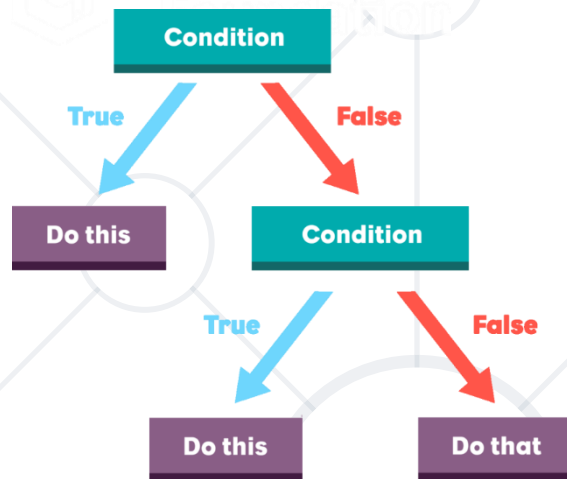


# По-сложни проверки

Вложени if конструкции и  
по-сложни логически условия



СофтУни  
Преподавателски екип



Имате въпрос?

sli.do

**#pb-march**

1. Преговор
2. Дебъгване
3. Проверки за съвпадение
4. Вложени условни конструкции
5. По-сложни проверки
  - Логическо "и", "или", отрицание и приоритет на условия
6. Решаване на изпитна задача





**Преговор**

1. Коя променлива е наименувана правилно?

`savedMoney`

`SavedMoney`

`спестениПари`

`spesteniPari4ki`

2. Каква стойност ще присвои променливата "isGreater":

```
let isGreater = (5 + 3) > (3 + 4);
```

true

false

8

15

3. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
if ("caseSensitive" == "CaseSensitive") {  
    console.log("Svetlin");  
} else {  
    console.log("Petar");  
}
```

Error

Svetlin

Petar

No output

4. Какъв ще е резултатът от изпълнението на следната програма:

```
console.log(123456 % 100 == 56);
```

true

false

56

Error



5. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
let role = "Administrator";  
if (role != "Administrator") {  
    console.log("No permission");  
} else {  
    console.log("Welcome");  
}
```

"Welcome"

No permission

No output

Compile time  
error



# Дебъгване

Прости операции с дебъгер

# Дебъгване

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



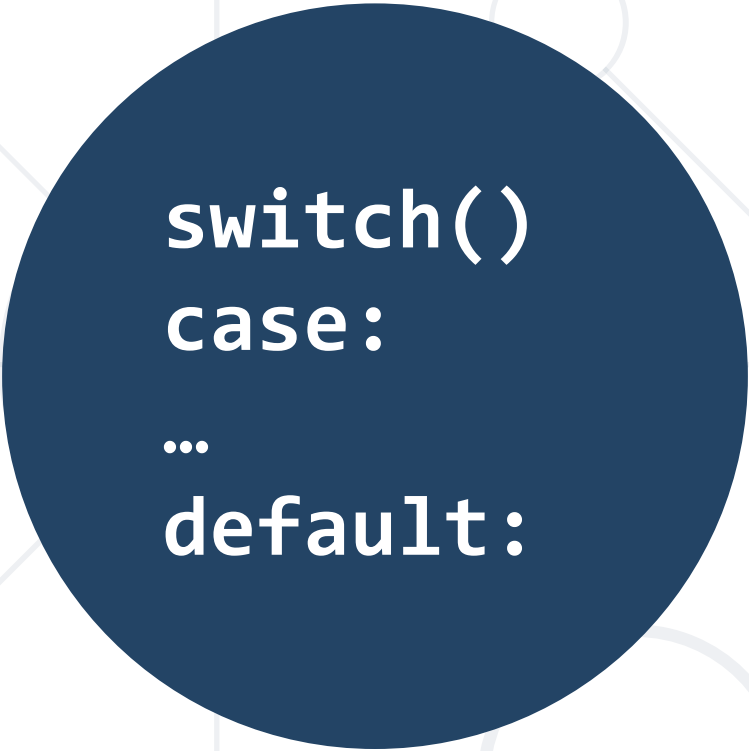
Breakpoint

```
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]





```
switch()  
case:  
...  
default:
```

**Условна конструкция switch-case**

# Условна конструкция switch-case

- Работи като поредица `if/else if/else if...`

```
switch (...) {  
  case ...:   
    // code  
    break;  
  case ...:   
    // code  
    break;  
  default:   
    // code  
    break;  
}
```

Изброяване на  
условия (стойности)  
за проверката

Условието в  
switch case  
е стойност

Код, който ще се изпълни,  
ако няма съвпадение с  
нико един случай

- Напишете програма, която:
  - Получава цяло число
  - Отпечатва на конзолата деня от седмицата с текст (на английски) според въведеното число [1...7]
  - Отпечатва на конзолата "**Error!**", ако числото не е в диапазона
- Примерен вход и изход:

1 → Monday

4 → Thursday

```
let day = Number(num);
switch (day)
{
    case 1:
        console.log("Monday"); break;
    case 2:
        console.log("Tuesday"); break;
    // TODO: check the other days
    case 7:
        console.log("Sunday"); break;
    default:
        console.log("Error!"); break;
}
```



# Множество случаи в switch-case

- Чрез **switch-case**, можем да изпълняваме един и същ код за множество условия

```
switch (...) {  
    case ... :  
    case ... :  
    case ... :  
        // code  
        break;  
    default:  
        // code  
        break;  
}
```

Кодът ще се изпълни за някое от трите условия в серията



```
if(condition)
```

**Вложени условни конструкции**

- Само при изпълнение на първото условие се преминава към вложената проверка

```
if (condition1) {  
    console.log("condition1 valid");  
    if (condition2)  
        console.log("condition2 valid");  
    else  
        console.log("condition2 not valid");  
}
```

Вложена if конструкция

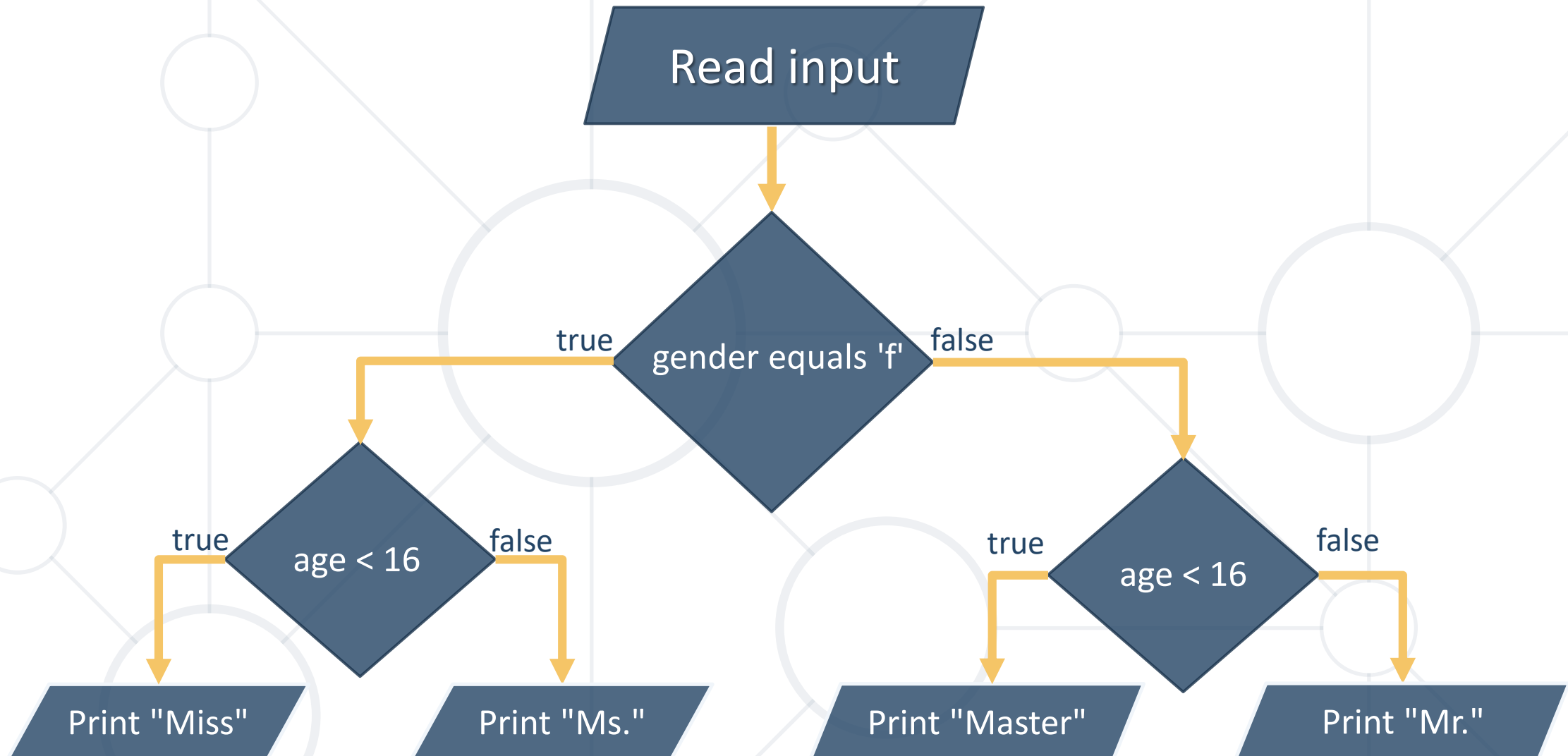
# Обръщение според възраст и пол – условие

- Напишете програма, която:
  - Получава като аргументи:
    - Възраст
    - Пол
  - Принтира обръщение според получените данни, както е показано на схемата (в следващия слайд)
- Примерен вход и изход:



12  
f → Miss

16  
m → Mr.



- Напишете програма, която:
  - Получава като аргументи:
    - Име на продукт
    - Град
    - Количество
  - Пресмята цената му спрямо таблицата:

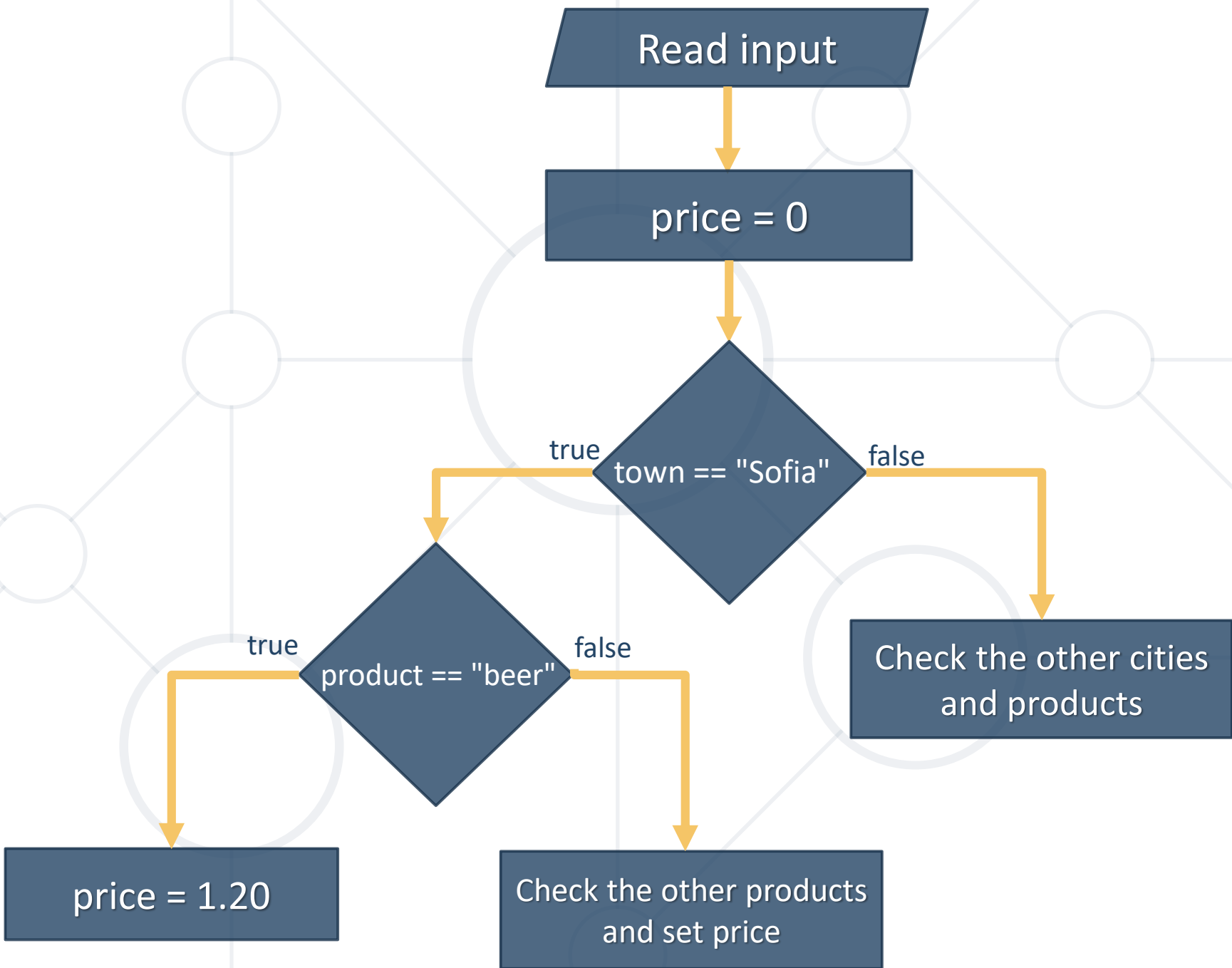


Град/продукт	coffee	water	beer	sweets	peanuts
Sofia	0.50	0.80	1.20	1.45	1.60
Plovdiv	0.40	0.70	1.15	1.30	1.50
Varna	0.45	0.70	1.10	1.35	1.55

# Квартално магазинче – условие (2)


- Примерен вход и изход:







```
function smallShop(productName, town, quantity){  
    quantity = Number(quantity);  
    let price = 0;  
    if (town == "Sofia") {  
        if (productName == "coffee")  
            price = quantity * 0.50;  
        // TODO: finish the checks for all the products..  
    }  
    else if (town == "Plovdiv")  
    else if (town == "Varna")  
    // TODO: check other two towns..  
}
```

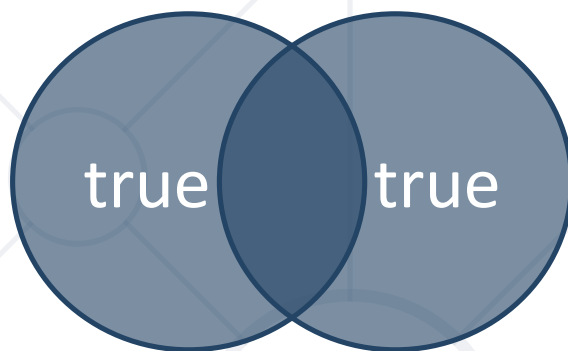
A background network diagram consisting of several light gray circles of varying sizes connected by thin gray lines. The circles are arranged in a non-uniform pattern, with some having more connections than others, creating a web-like structure.

```
if ()  
else if()  
...  
else
```

**По-сложни проверки**  
**Булеви оператори**

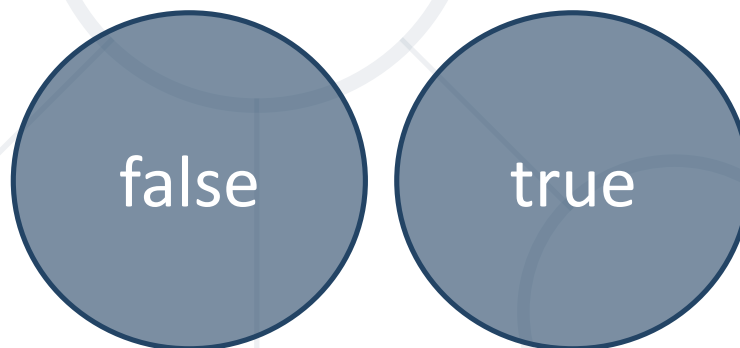
- Оператори, които комбинират или изключват условия
- Връщат булеви резултат (true или false)

"&&" - И



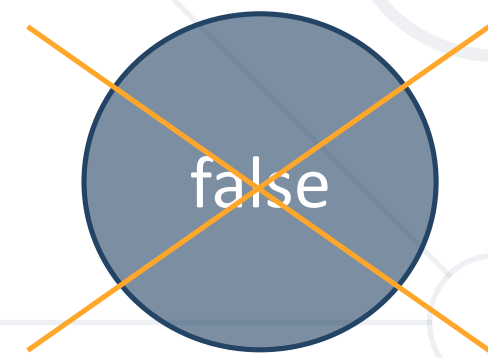
Вярност на двете условия

"||" - ИЛИ



Вярност на  
едното или на другото  
условие

"!" - ОТРИЦАНИЕ



Отрицание на условие

- Проверява изпълнението на няколко условия едновременно
- Пример: проверка дали число е едновременно:
  - по-голямо от 5 и по-малко от 10
  - четно

&&

```
a = Number(a);  
if ((a > 5 && a < 10) && a % 2 == 0) ...
```

- Напишете програма, която:
  - Проверява дали въведеното число от потребителя е в интервала  $[-100, 100]$  и е различно от 0
  - Извежда "Yes", ако е в интервала и различно от 0, или "No" ако е извън тях.
- Примерен вход и изход:



# Число в интервала - решение

```
function numberInRange(number){  
    number = Number(number);  
  
    if (number >= -100 && number <= 100 && number != 0) {  
        console.log("Yes");  
    } else {  
        console.log("No");  
    }  
}
```

- Проверява, дали е изпълнено поне едно измежду няколко условия
- Пример: проверка дали получената дума е:
  - "Example" или "Demo"

```
if (word == "Example" || word == "Demo") ...
```

- Напишете програма, която:
  - Проверява дали полученият аргумент е плод или зеленчук измежду изброените:
    - Плодове: banana, apple, kiwi, cherry, lemon, grapes
    - Зеленчуци: tomato, cucumber, pepper, carrot
  - Извежда **"vegetable"**, **"fruit"** или **"unknown"**
- Примерен вход и изход:

lemon



fruit

carrot



vegetable

java



unknown



# Плод или зеленчук - решение

```
function fruitOrVegetable(food){  
  if (food == "banana" || food == "apple" || food == "kiwi" ||  
      food == "cherry" || food == "lemon" || food == "grapes")  
    console.log("fruit");  
  else if (food == "tomato" || food == "cucumber" ||  
           food == "pepper" || food == "carrot")  
    console.log("vegetable");  
  else  
    console.log("unknown");  
}
```

- Чрез скоби ( ) можем да приоритизираме условия

```
let a = 50;
let b = 200;
let c = 300;

if ((a >= 100 && b <= 200) || (c + b >= 300 && c <= 400)) {
  console.log("Yes"); // Yes
}

if (a >= 100 && (b <= 200 || c + b >= 300) && c <= 400) {
  console.log("Yes"); // No output
}
```

- Проверява дали **не е изпълнено** дадено условие
- Пример:
  - Проверка дали число е по-голямо от 10 и е четно:

```
function invalidNumber(number){  
    number = Number(number);  
    let isValid = (number > 10) && (number % 2 == 0);  
    if (!isValid) {  
        console.log("Invalid");  
    } else {  
        console.log("Valid");  
    }  
}
```





**По-сложни проверки**  
**Решаване на задачи в клас (лаб)**

- Напишете програма, която:
  - Получава като аргументи:
    - Продукт
    - Ден
    - Количество
  - Извежда сумата, която трябва да се заплати според деня и продукта



# Магазин за плодове - условие (2)

- В работни дни продава на следните цени:

плод	banana	apple	orange	grapefruit	kiwi	pineapple	grapes
цена	2.50	1.20	0.85	1.45	2.70	5.50	3.85

- В почивни дни цените са по-високи:

плод	banana	apple	orange	grapefruit	kiwi	pineapple	grapes
цена	2.70	1.25	0.90	1.60	3.00	5.60	4.20

- Примерен вход и изход:

apple  
Tuesday  
2



2.40

orange  
Sunday  
3



2.70

```
if (day == "saturday" || day == "sunday"){  
    if (fruit == "banana") price = 2.70;  
    else if (fruit == "apple") price = 1.25;  
    // TODO: check the other fruits  
}  
else if (day == "monday" || day == "tuesday" ||  
        day == "wednesday" || day == "thursday" ||  
        day == "friday"){  
    if (fruit == "banana") price = 2.50;} // TODO ...
```

- Напишете програма, която:
  - Получава:
    - Град
    - Обем на продажби (реално число)
  - Изчислява комисионната, която дадена фирма дава на търговците според града и обема на продажбите
  - Извежда стойността на комисионната, закръглена до 2 цифри след десетичната запетая





# Търговски комисионни – условие (2)

Град / цена	$0 \leq s \leq 500$	$500 < s \leq 1\,000$	$1\,000 < s \leq 10\,000$	$s > 10\,000$
Sofia	5%	7%	8%	12%
Varna	4.5%	7.5%	10%	13%
Plovdiv	5.5%	8%	12%	14.5%

- Примерен вход и изход:



```
function tradeCommissions(town, sales){
  let commission = -1.0;
  if (town == "Sofia")
    if (sales >= 0 && sales <= 500) comission = 0.05;
    else if (sales > 500 && sales <= 1000) comission = 0.07;
  // TODO: check the other price ranges...
  else if (town == "Varna") // TODO: check the price ranges...
  else if (town == "Plovdiv") // TODO: check the price ranges...

  if (commission >= 0)
    console.log(`${(sales * commission).toFixed(2)}`);
  else
    console.log("error");
}
```

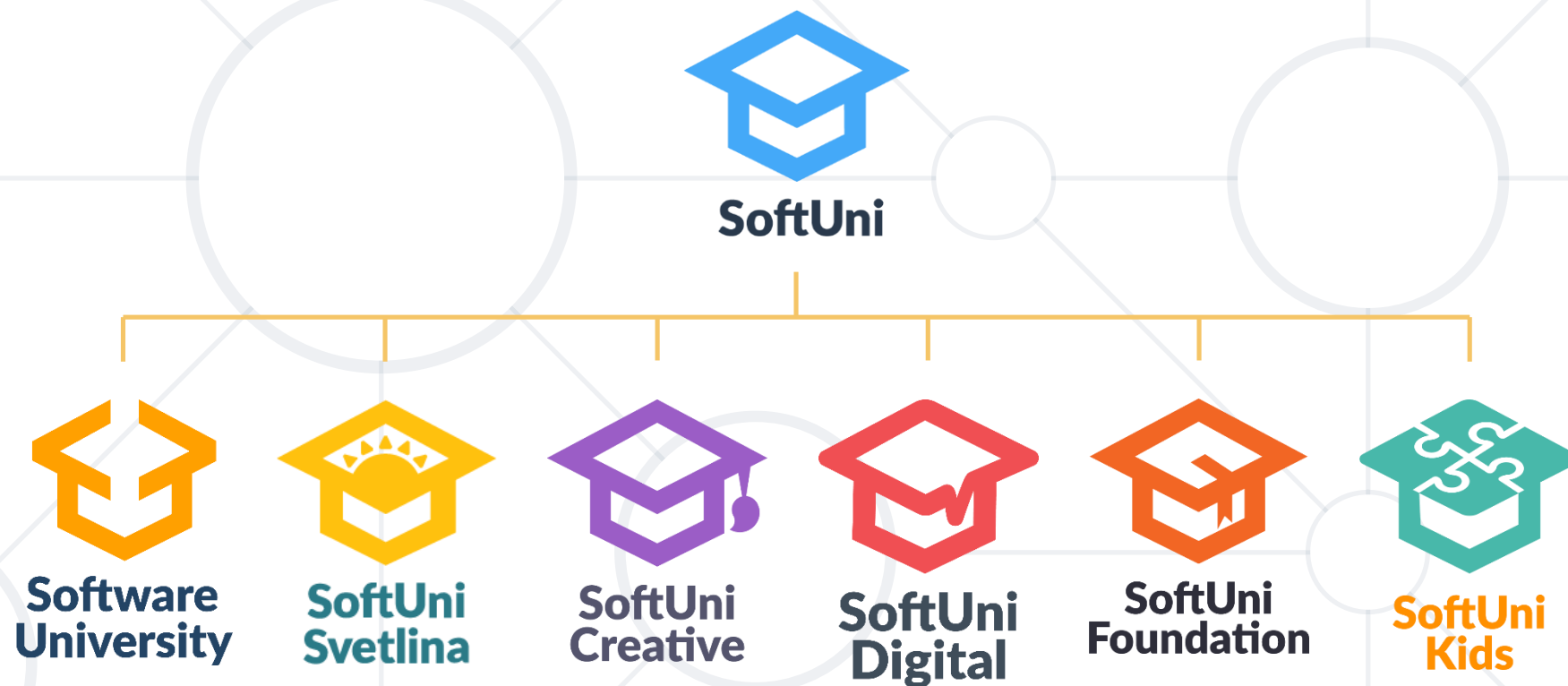


**Вложени условни конструкции**  
**Решаване на задачи в клас (лаб)**

- Дебъгване
- Конструкцията **switch-case**
- Вложени условни конструкции
- По-сложни проверки с **&&**, **||**, **!** и **()**



# Въпроси?



# SoftUni Diamond Partners



**XS**software



**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



OneBit  
SOFTWARE



WORLD  
OF  
MYTHS

- Настоящият курс (слайдове, примери, видео, задачи и др.) с е 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-SA](#)



# Обучения в СофтУни

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



**Software  
University**

