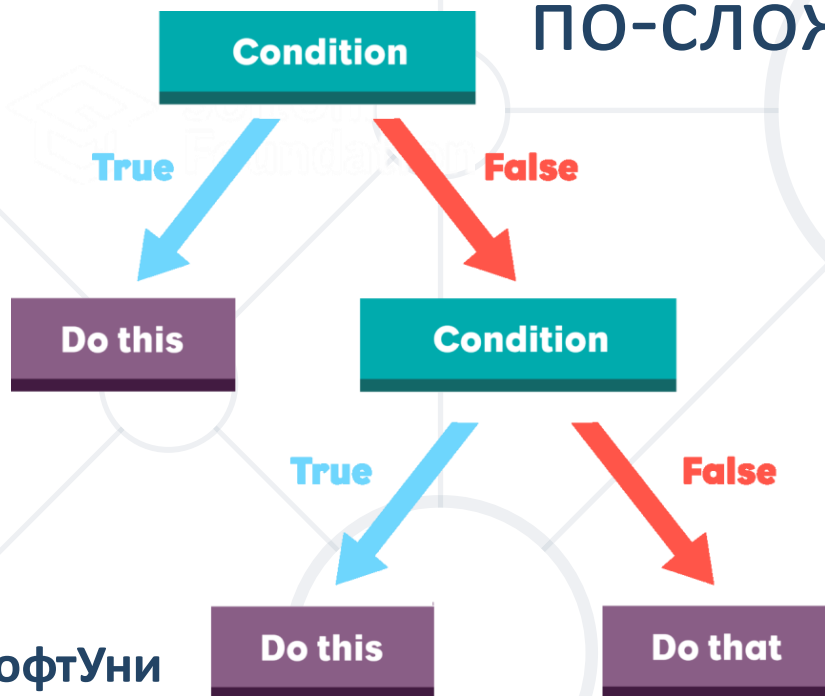


По-сложни проверки

Вложени If конструкции и
по-сложни логически условия



СофтУни
Преподавателски екип



SoftUni



<https://softuni.bg>

Софтуерен университет

1. Преговор
2. Условна конструкция **switch - case**
3. Вложени условни конструкции
4. Логически оператори
 - Логически оператори **"&&"**, **"||"**, **"!"**
 - Приоритет на условия





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");  
}
```

Svetlin

Error

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**

A background network diagram consisting of a grid of light gray lines intersecting at various points. At these intersections, there are several light gray circles of different sizes. A large, solid dark blue circle is positioned in the upper-middle part of the image, containing white text. Below this circle, the title 'Условна конструкция Switch-case' is written in a bold, dark blue font.

```
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
нито един случай

- Напишете функция, 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

5 → Friday

2 → Tuesday

```
function dayOfWeek(number) {  
  switch (number) {  
    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;  
  }  
}  
solve(1); // Monday
```

- 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
ако някое от трите
условия в серията е
вярно

- Напишете функция, която:
 - Получава ден от седмицата (**текст**)
 - Ако денят е работен - отпечатва на конзолата "**Working day**"
 - Ако денят е почивен - отпечатва на конзолата "**Weekend**"
 - Ако се въведе текст различен от ден от седмицата - отпечатва на конзолата "**Error**"
- Примерен вход и изход:

Monday



Working day

Sunday



Weekend

Почивен или работен ден - решение

```
function typeDay(day) {  
  switch (day) {  
    case "Monday":  
    case "Tuesday":  
      // TODO  
      console.log("Working day"); break;  
    case "Saturday":  
    case "Sunday":  
      console.log("Weekend"); break;  
    default:  
      console.log("Error"); break;  
  }  
}  
solve("Monday"); // Working day
```

- Напишете функция, която:
 - Получава като аргумент **текст** и проверява дали 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


```
switch (food) {  
  case "banana":  
    // TODO for all the fruits  
  case "grapes":  
    console.log("fruit"); break;  
  case "tomato":  
    // TODO for all the vegetables  
  case "carrot":  
    console.log("vegetable"); break;  
  default:  
    console.log("unknown"); break;  
}
```



```
{ if else }
```

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 функция, която получава като аргументи:
 - Възраст и пол
 - Принтира обръщение според въведените данни, както е показано на схемата (в следващия слайд)
- Примерен вход и изход:

12
f



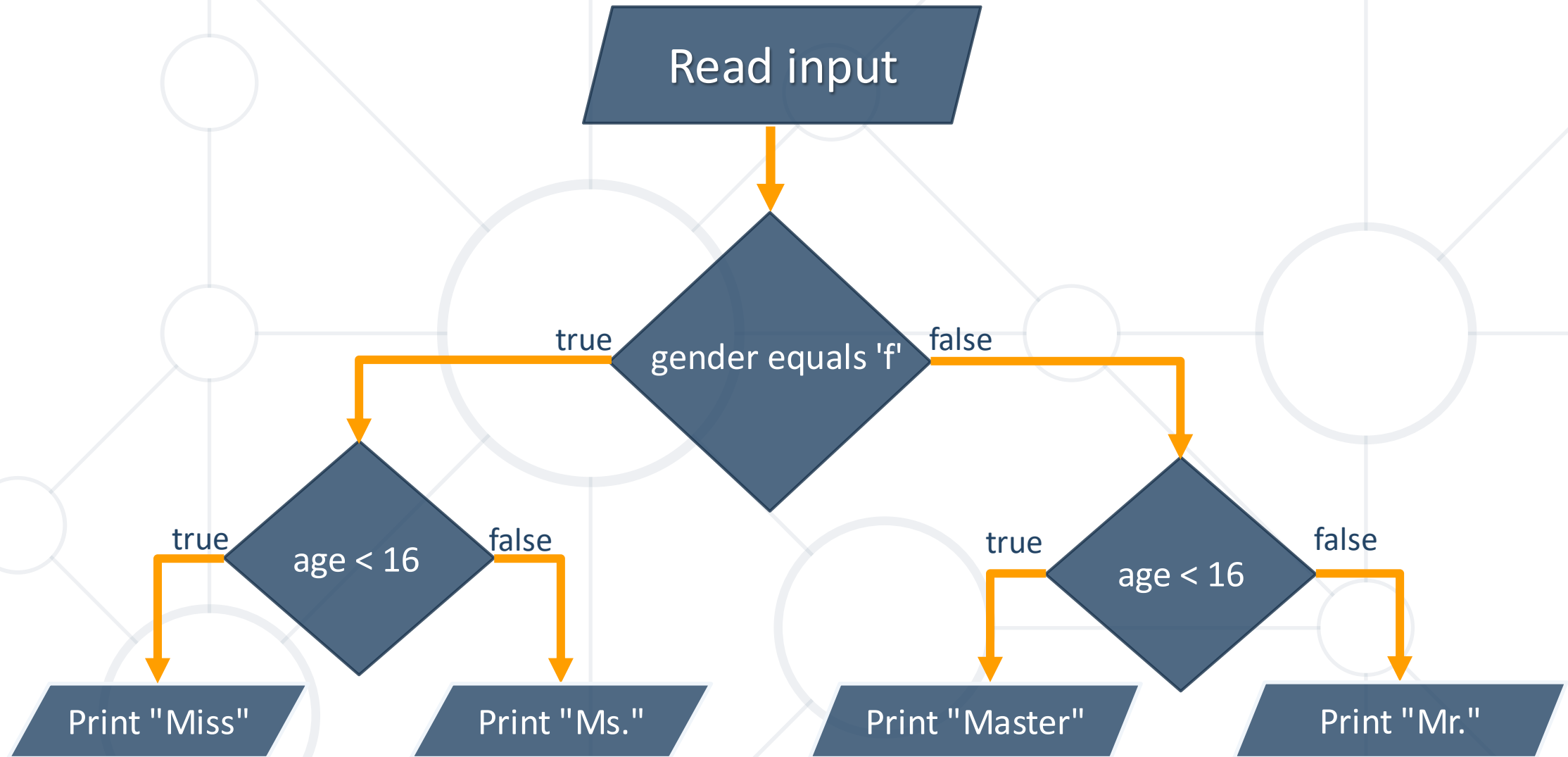
Miss

16
m



Mr.





```
if (gender == "f") {  
  if (age >= 16) {  
    console.log("Ms.");  
  }  
  else {  
    console.log("Miss");  
  }  
}  
else {  
  //TODO: check other titles - "Mr.", "Master"  
}
```

- Напишете функция, която получава като аргументи:

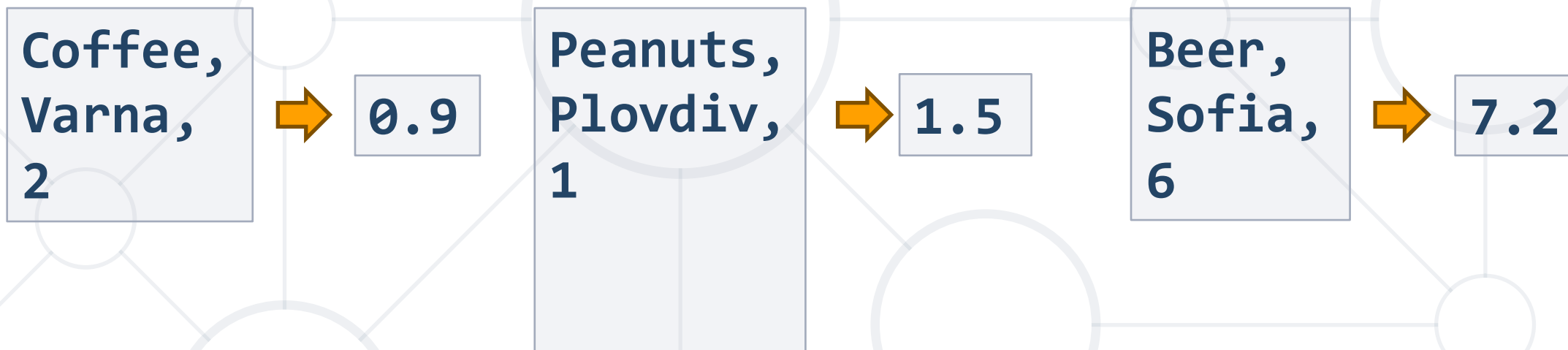
- Име на продукт
- Град
- Количество

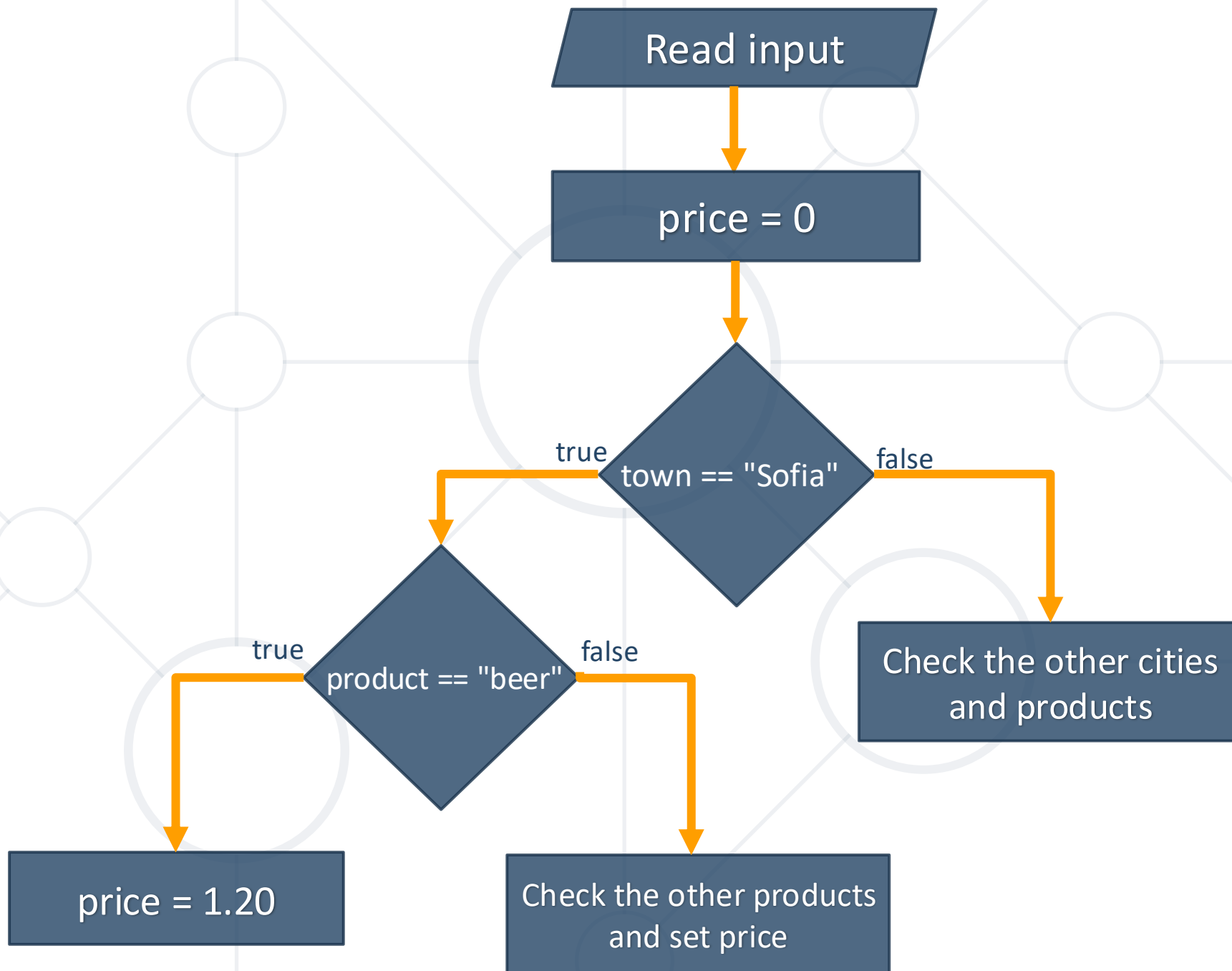
- Пресмята цената му спрямо таблицата:

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



- Примерен вход и изход:





```
function solve(product, town, quantity) {  
  let price = 0;  
  if (town == "Sofia") {  
    if (product == "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..  
}
```

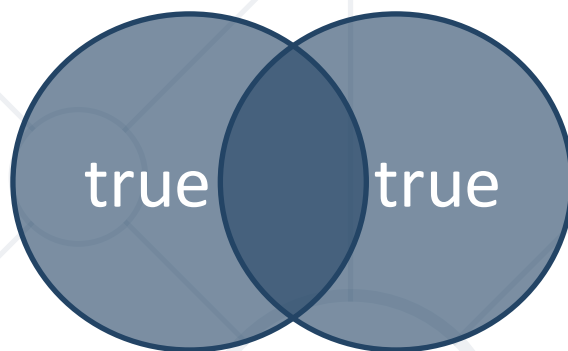
A background network diagram consisting of a grid of light gray lines intersecting at various points. Some of these intersections are marked with small, empty light gray circles. A large, solid dark blue circle is centered in the upper half of the image, containing the logical operators. Below this circle, the title is written in a bold, dark blue font.

**&&
||
!**

Логически оператори

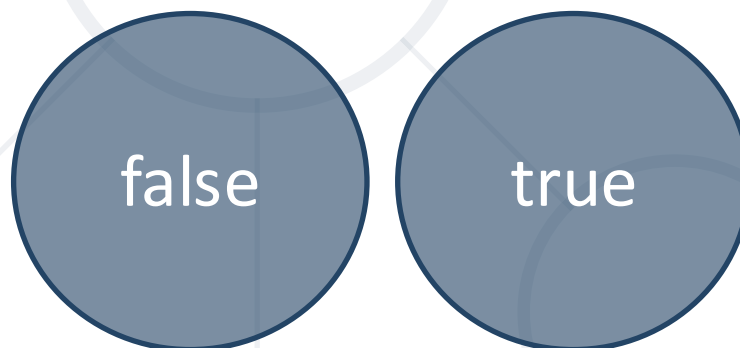
- Оператори, които комбинират или изключват условия
- Връщат булев резултат (true или false)

"&&" - И



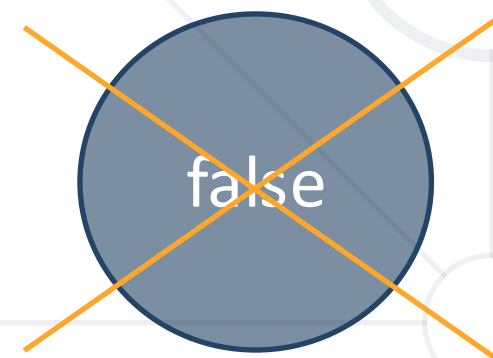
Вярност на двете условия

"||" - 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

&&

```
function solve(a) {  
    if (a > 5 && a < 10 && a % 2 == 0)...  
}
```

- Вложенные проверки

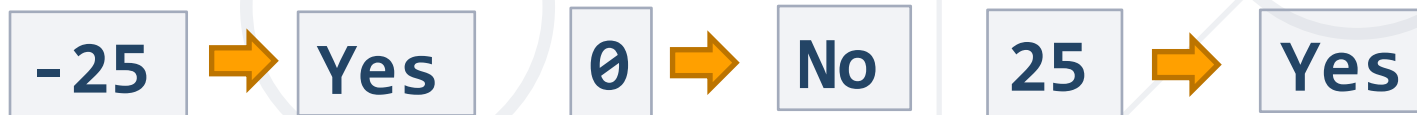
```
if (a > 5) {  
    if (a < 10) {  
        if (a % 2 == 0) {  
            }  
        }  
    }  
}
```

- Логический оператор **&&**

```
if (a > 5 && a < 10  
    && a % 2 == 0) {  
}
```



- Напишете функция, която:
 - Приема като аргумент число и проверява дали 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 или е 0.
- Примерен вход и изход:



Число в интервала – решение

```
function numberInRange (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"

```
function solve(word) {  
    if (word == "Example" || word == "Demo")...  
}
```

- Проверки

```
if (word == "Example"){  
}  
else if (word == "Demo"){  
}
```

- Логически оператор ||

```
if (word == "Example" ||  
    word == "Demo"){  
}
```



- Напишете функция, която:
 - Приема за аргумент - ден от седмицата (**текст**)
 - Отпечатва цената на билет за кино според деня от седмицата

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12	12	14	14	12	16	16

- Примерен вход и изход:

Monday → 12

Sunday → 16

```
function cinemaTicket (day) {  
  if (day == "Monday" || day == "Tuesday" || day == "Friday") {  
    console.log(12);  
  } else if (day == "Wednesday" || day == "Thursday") {  
    console.log(14);  
  }  
  // TODO for Saturday and Sunday  
}
```

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

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



- Напишете функция, която:
 - Приема за аргумент цяло число
 - Числото е валидно ако е в интервала [100...200] или е 0
 - Ако числото е невалидно да се отпечата на конзолата "invalid", в противен случай да не се отпечата нищо
- Примерен вход и изход:

75 → invalid

150 → (няма изход)

```
function invalidNumber (n) {  
  let isValid = n >= 100 && n <= 200 || n == 0;  
  if (!isValid) {  
    console.log("invalid");  
  }  
}
```

- Чрез скоби **()** можем да приоритизираме условия

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

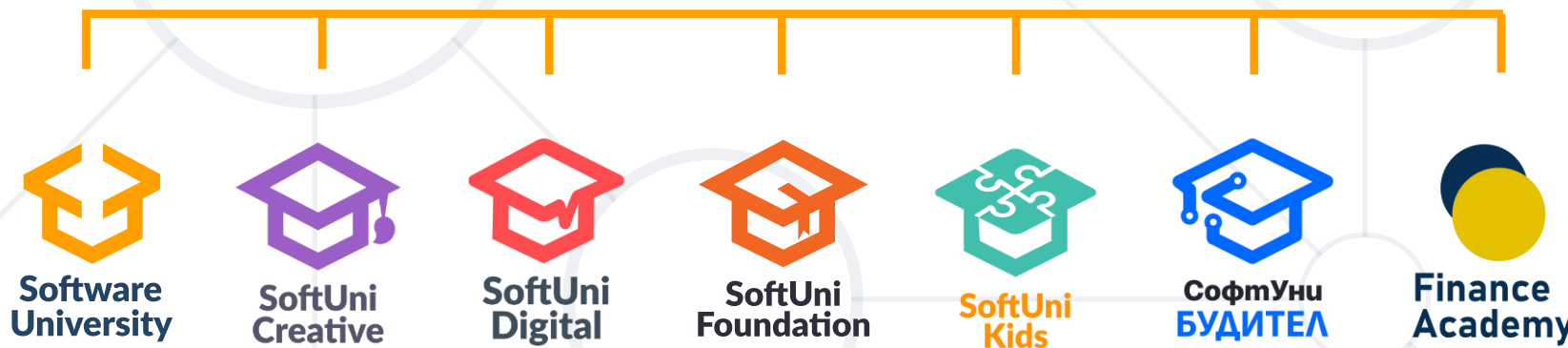
if ((a >= 100 && b <= 200) || (c + b >= 300 && c <= 400)) {
  console.log("Yes"); // Yes
}

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


- Условна конструкция - **switch-case**
- Вложени условни конструкции
- Логически оператори - **&&, ||, !**
 - Приоритет на условия – **()**



Въпроси?



- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>



- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity

