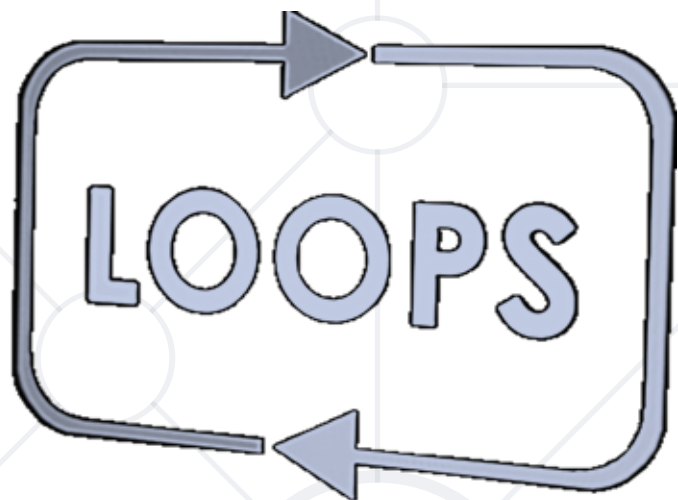


Повторения (цикли)

Прости повторения с For-цикъл



СофтУни

Преподавателски екип



SoftUni



Софтуерен университет

<https://softuni.bg>

1. Преговор
2. Увеличаване и намаляване на стойността на променлива
3. Повторения на блокове код
4. Работа с по-сложни for-цикли
5. Работа с текст
6. Техники за използване на for-цикли





Преговор

1. Каква ще е стойността на променливата **a** след изпълнението на следната програма:

```
let a = 5;  
switch (a){  
  case 5:  
  case 6:  
    a = a + 1;  
    break;  
  default:  
    a = a + 2;  
    break;  
}
```

0

5

6

7

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

```
console.log(!(5 === 5) && (4 + 1 === 5));
```

True

False

**Runtime
error**

**Compile time
error**

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

```
console.log(!(3 === 3) || (3 === 5));
```

True

Runtime
error

False

Compile time
error

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

```
console.log(!(3 > 5) || (1 === 1));
```

**Compile time
error**

**Runtime
error**

False

True

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

```
let number = 101;  
if (number >= 1)  
  console.log("Larger than 1");  
if (number <= 101)  
  console.log("Less than 101");  
  console.log("Equal to 101");
```

Larger than 1

Less than 101

Equal to 101

Compile
time error

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

```
let role = "Administrator";  
let password = "SoftUni";  
if (role === "SoftUni") {  
    if (password === "SoftUni") {  
        console.log("Welcome!");  
    }  
}
```

"Welcome!"

Runtime error

No output

Compile time error



**Увеличаване и намаляване на стойността
на променливи**

- **Инкрементиране** - увеличаването на стойността на дадена променлива
 - Извършва се чрез оператори за инкрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

Пример	Име	Резултат
<code>++a</code>	Пре -инкрементация	Увеличава стойността с единица и връща a
<code>a++</code>	Пост -инкрементация	Връща a и увеличава стойността с единица

- **Пре**-инкрементация

```
let a = 1;  
console.log(++a); // 2  
console.log(a);  // 2
```

Стойността на променливата a се увеличава с 1 и след това се принтира

- **Пост**-инкрементация

```
let a = 1;  
console.log(a++); // 1  
console.log(a);  // 2
```

Първо се принтира променливата a и след това се увеличава с 1

- **Декрементиране** – намаляването на стойността на дадена променлива
 - Извършва се чрез оператори за декрементиране: **префиксни** и **постфиксни**
 - Извършва се само върху променливи, които имат числена стойност

Пример	Име	Резултат
--a	Пре -декрементация	Намалява стойността с единица и връща a
a--	Пост -декрементация	Връща a и намалява стойността с единица

- **Пре**-декрементация

```
let a = 1;  
console.log(--a); // 0  
console.log(a);  // 0
```

Стойността на променливата а се намалява с 1 и след това се принтира

- **Пост**-декрементация

```
let a = 1;  
console.log(a--); // 1  
console.log(a);  // 0
```

Първо се принтира променливата а и след това се намалява с 1



Повторения на блокове код

Конструкция за For-цикъл

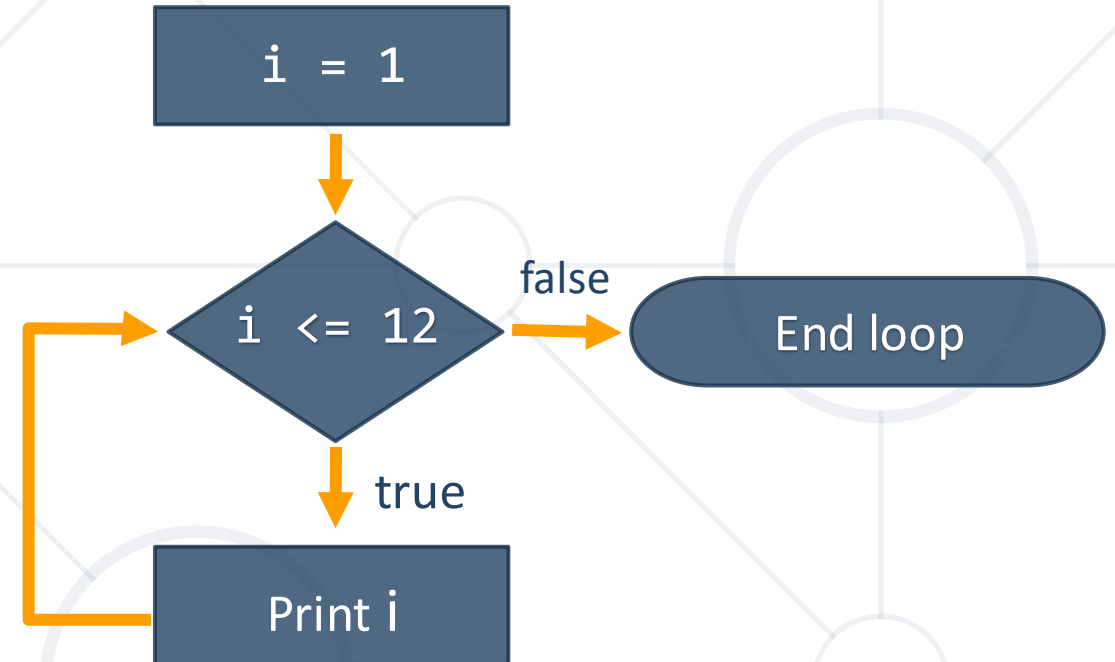
Какво е цикъл?

- Често ни се налага да **повтаряме** едно и също действие **многократно**
- Когато сме абитуриенти броим до 12



Какво е цикъл? (2)

- Циклите в програмирането ни позволяват да повтаряме **едни и същи действия** определен брой пъти:



```
for (let i = 1; i <= 12; i += 1) {  
  console.log(i);  
}
```

For-цикъл – конструкция

- Можем да повтаряме действия до определен момент чрез **for**-цикли

Ключова дума за
конструкцията

Начална
стойност

Крайна
стойност

```
for (let i = 1; i <= 12; i += 1) {  
  console.log(i);  
}
```

Стъпка

Тяло на цикъла:
блок от код за повторение



Работа с по-сложни For-цикли

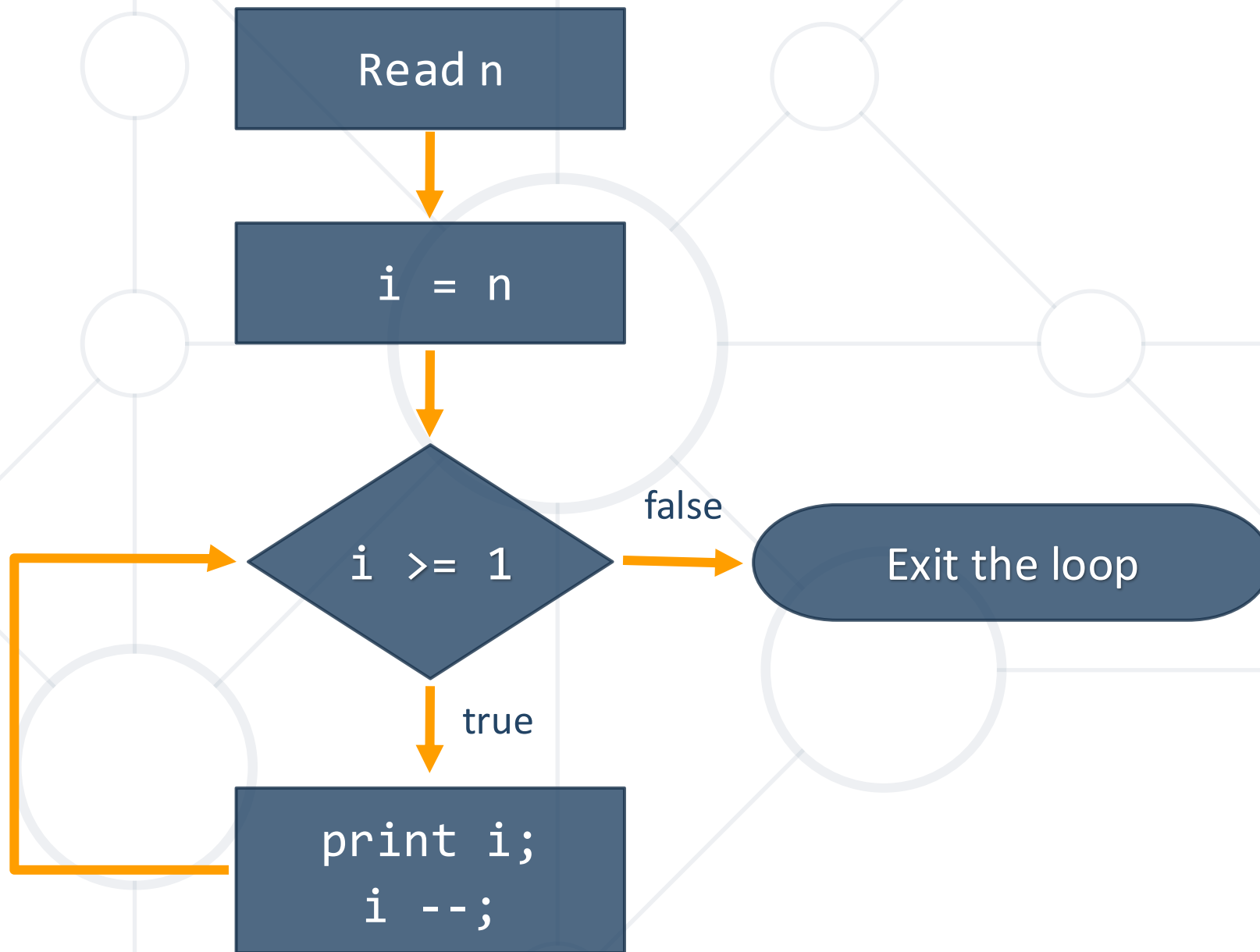
Цикли със стъпка

Числата от N до 1 в обратен ред – условие

- Напишете функция, която:
 - Получава цяло положително число **n**
 - Отпечатва числата от **n** до **1** в обратен ред (стъпка -1)
- Примерен вход и изход:

100 → 100, 99, 98, ..., 3, 2, 1

8 → 8, 7, 6, 5, 4, 3, 2, 1



Числата от N до 1 в обратен ред – решение

```
function nToOneReverse (n){  
  for (let i = n; i >= 1; i--) {  
    console.log(i);  
  }  
}
```

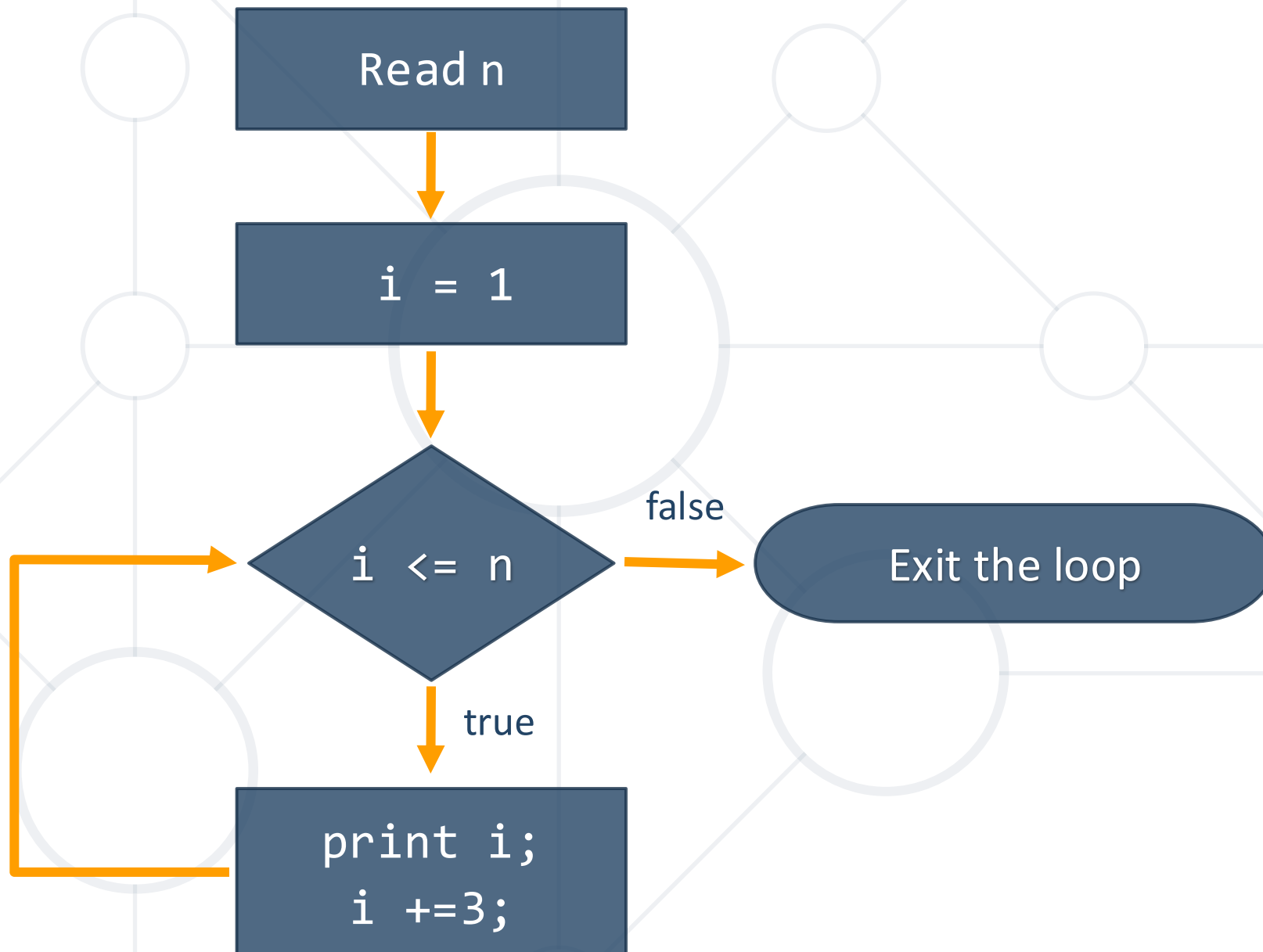
Обърнато условие:
 $i \geq 1$

Намаляваща стъпка: -1

Числата от 1 до N през 3 – условие

- Напишете функция, която:
 - Получава цяло число **n**
 - Отпечатва числата от **1** до **n** със стъпка **3**
- Примерен вход и изход:





Числата от 1 до N през 3 – решение

```
function oneToN (n){  
  for (let i = 1; i <= n; i += 3){  
    console.log(i);  
  }  
}
```

Задаване на стъпка 3

Четни степени на 2 – условие

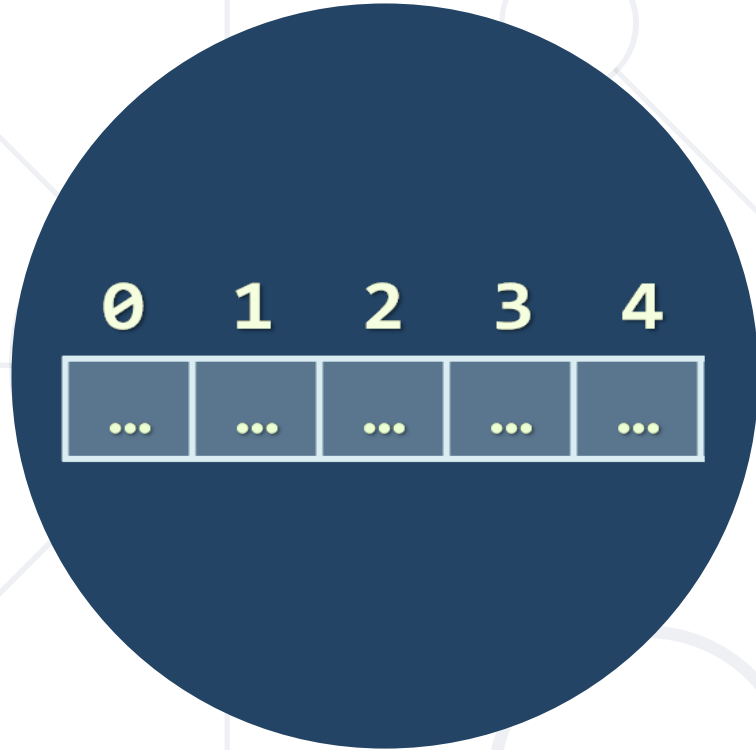
- Напишете функция, която:
 - Получава цяло число **n**
 - Отпечатва четните степени на **2** до **2ⁿ**: $2^0, 2^2, 2^4, 2^8, \dots, 2^n$
- Примерен вход и изход:

10 → 1, 4, 16 , ..., 1024

7 → 1, 4, 16 , ..., 64

```
function evenDegrees (n){  
  let num = 1;  
  for (let i = 0; i <= n; i += 2) {  
    console.log(num);  
    num = num * 2 * 2;  
  }  
}
```

Ползваме
стъпка 2



Масиви – четене от масив

Какво е масив?

В програмирането **масив** е **поредица от елементи**



- Можем да запазваме **много стойности** в една променлива
- Елементите са номерирани от **0** до **length-1**
- Масивите имат **дължина** (**Array.length**)

- Както от подаден текст можем да прочетем само една буква, така можем да прочетем само един елемент от даден масив.

```
function solve (input){  
    console.log(input[0]); //a  
    console.log(input[1]); //b  
    console.log(input[2]); //c  
}  
solve([ "a", "b", "c" ])
```



Работа с текст

- Можем да вземем дължината на текст

```
let text = 'SoftUni';  
let length = text.length; // 7
```

- Можем да вземем символ от текст по индекс (използване на индексирал оператор)

```
let text = 'SoftUni';  
let letter = text[4]; // u
```


- Напишете функция, която
 - получава текст(**стринг**)
 - печата всеки **СИМВОЛ** от текста на отделен ред
- Примерен вход и изход:



Взимаме
дължината на
текста

```
function streamOfCharacters(text) {  
  for (let i = 0; i < text.length; i++) {  
    console.log(text[i]);  
  }  
}
```

Взимаме всеки символ
по индекс i

- Можем да вземем символ от текст по индекс, като използваме метода **charAt(index)** – където index е число, от което ще вземем съответната буква

```
let text = 'SoftUni';  
let letter = text.charAt(4); // u
```

Сумиране на гласни букви – условие

- Напишете функция, която:
 - Получава текст
 - Извежда сумата на гласните букви според таблицата по-долу:

а	е	и	о	и
1	2	3	4	5

- Примерен вход и изход:

hello



6

(e+o = 2+4 = 6)

hi



3

(i = 3)

bamboo



9

(a+o+o = 1+4+4 = 9)

beer



4

(e+e = 2+2 = 4)

Сумиране на гласни букви – решение

```
function vowelsSum(text){  
  let sum = 0;  
  for (let i = 0; i < text.length; i++) {  
    switch (text.charAt(i)) {  
      case 'a': sum += 1; break;  
      case 'e': sum += 2; break;  
      // TODO: Add cases for the other vowels.  
    }  
  }  
  console.log("Vowels sum = " + sum);  
}
```



Техники за използване на For-цикли

Задачи с цикли

- Напишете функция, която:
 - Получава цяло число **n**, под формата на **текст**
 - Сумира отделните цифри в него
 - Извежда пресметнатата сума
- Примерен вход и изход:

"1234"



The sum of the digits is:10

"564891"



The sum of the digits is:33

Read input

sum = 0

$i \leq \text{input.length}$

false

Print the sum

true

Read a number;
Add it to the sum




```
function sumDigits(numberAsString) {  
    let sum = 0;  
  
    for (let i = 0; i < numberAsString.length; i++){  
        sum += Number(numberAsString[i]);  
    }  
    console.log(`The sum of the digits is:${sum}`);  
}
```

- Напишете функция, която:
 - Получава два аргумента **(n, m)** – цели числа
 - Открива всички числа в диапазона между двете входни, **които се делят на 9**
 - Принтира на първия ред резултата от сумата на числата
 - Принтира на втория ред отговарящите на условието числа, всяко от които на нов ред

Числа, които се делят на 9 – решение

```
function divisibleByNine(n, m) {  
  let sum = 0;  
  let outputNumbers = '';  
  for (let i = n; i < m; i++) {  
    if (i % 9 === 0) {  
      // TODO: Add the number to the sum.  
      // TODO: Add the number to the outputNumbers.  
    }  
  }  
  console.log(// TODO: );  
  console.log(// TODO: );  
}
```

- Повторение на блок код с **for**-цикъл
- Цикли със стъпка
 - Цикли с увеличаваща стъпка
 - Цикли с намаляваща стъпка
- Вземане на символ по индекс от текст



Въпроси?



SoftUni



Software
University



SoftUni
Creative



SoftUni
Digital



SoftUni
Foundation



SoftUni
Kids



СофтУни
БУДИТЕЛ



Finance
Academy

- 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

