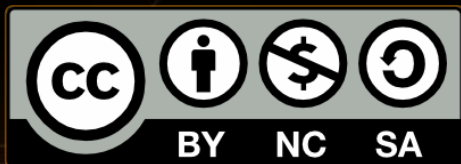


Повторения (цикли)

Прости повторения с for-цикъл



СофтУни

трейнърски екип

Софтуерен университет

<http://softuni.bg>



Повторения

LOOPS

Have a Question?

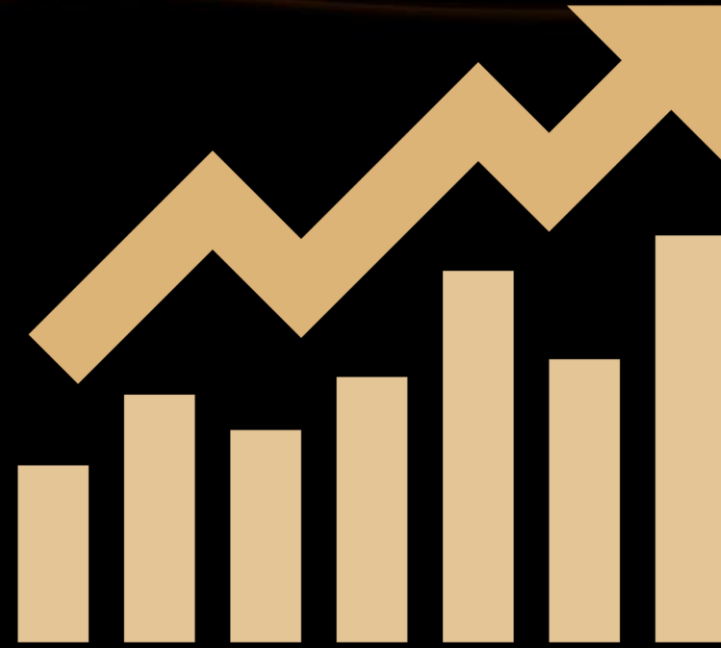
sli.do

#TODO

Съдържание

1. Увеличаване и намаляване на стойността на променливи
2. For-цикъл
 - Конструкция





**Увеличаване и намаляване на
стойността на променливи**

- **Инкрементиране** – увеличаване на стойността на дадена променлива
 - Извършва се чрез оператори за инкрементиране – **префиксни** и **постфиксни**

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

- Извършва се **само** върху променливи, които имат **числена** стойност

Увеличаване (2)

■ Пре-инкрементация

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

■ Пост-инкрементация

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

Намаляване

- **Декрементиране** – намаляването на стойността на дадена променлива
 - Извършва се чрез оператори за декрементиране – **префиксни** и **постфиксни**

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

- Извършва се **само** върху променливи, които имат **числена** стойност

Намаляване (2)

■ Пре-декрементация

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

■ Пост-декрементация

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


LOOPS

```
function numbers1To100() {  
  for (let i = 1; i <= 100; i++) {  
    console.log(i);  
  }  
}
```

Повторения на блокове код

Конструкция за цикъл for

Повторения (цикли)

- В програмирането често се налага да изпълним блок с команди няколко пъти
 - За целта използваме цикли

Ключова дума
за **for** цикъл

Начална
стойност

Крайна
стойност

Стъпка на
индекса (i)

```
for (let i = 1; i <= 10; i++) {  
  console.log("i = " + i);  
}
```

Тяло на цикъла: блок
команди за изпълнение

Числа от 1 до 100

- Напишете програма, която:
 - Извежда числата [1, 100], всяко на нов ред
- Решение:

```
function numbers1To100() {  
  for (let i = 1; i <= 100; i++) {  
    console.log(i);  
  }  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#0>

Числа до 1000, завършващи на 7

- Напишете програма, която:
 - Извежда числата [1, 1000], които завършват на 7
- Решение:

```
function numbersIn7() {  
  for (let i = 0; i <= 1000; i++) {  
    if (i % 10 == 7) {  
      console.log(i);  
    }  
  }  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#1>

ASCII таблица

- Символите, които използваме се представят като числа
 - Поместени са в **ASCII таблица**
- Примери (знак и неговата ASCII стойност):

a	→	97
---	---	----

A	→	65
---	---	----

@	→	64
---	---	----

+	→	43
---	---	----

[Пълна информация за ASCII](#)

Всички латински букви - условие

- Напишете програма, която:
 - Извежда буквите от латинската азбука: **[a, z]**
- Решение:

```
function latinLetters() {  
    for (let i = 97; i <= 122; i++) {  
        console.log(String.fromCharCode(i));  
    }  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#2>

Сумиране на числа - условие

- Напишете програма, която:
 - Прочита цяло число(**n**) от потребителя
 - Прочита **n**-на брой числа и ги **сумира**
 - Извежда пресметнатата сума
 - Примерен вход и изход:

2
10
20



30

3
-10
-20
-30



-60

4
45
-20
7
11



43

Сумиране на числа - решение

```
function sumNumbers(args) {  
    let n = Number(args[0]);  
    let sum = 0;  
    for (let i = 1; i <= n; i++) {  
        sum = sum + Number(args[i]);  
    }  
    console.log(sum);  
}
```

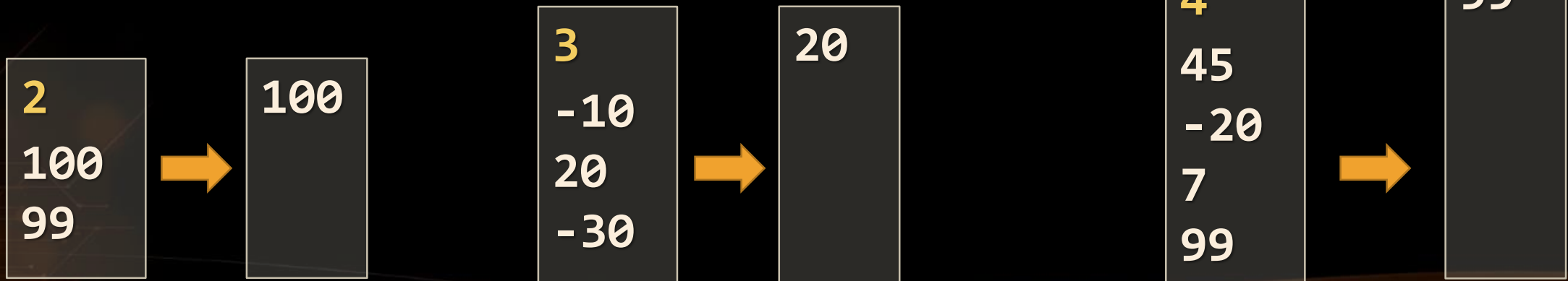
Масив от аргументи

Елемент от масива с
аргументи

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#3>

Най-голямо число - условие

- Напишете програма, която:
 - Прочита цяло число(**n**) от потребителя
 - Прочита **n**-на брой числа
 - Намира **най-голямото** измежду тях
- Примерен вход и изход:



Най-голямо число - решение

```
function maxNumber(args) {  
    let n = Number(args[0]);  
    let max = Number.NEGATIVE_INFINITY;  
    for (var i = 1; i <= n; i++) {  
        let num = Number(args[i]);  
        if (num > max) {  
            max = num;  
        }  
    }  
    console.log("max = " + max);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#4>

Най-малко число - условие

- Напишете програма, която:
 - Прочита цяло число(**n**) от потребителя
 - Прочита **n**-на брой числа
 - Намира **най-малкото** измежду тях
- Примерен вход и изход:



Най-малко число - решение

```
function maxNumber(args) {  
    let n = Number(args[0]);  
    let max = Number.POSITIVE_INFINITY;  
  
    //TODO: Use logic similar to "Biggest  
number".  
}
```

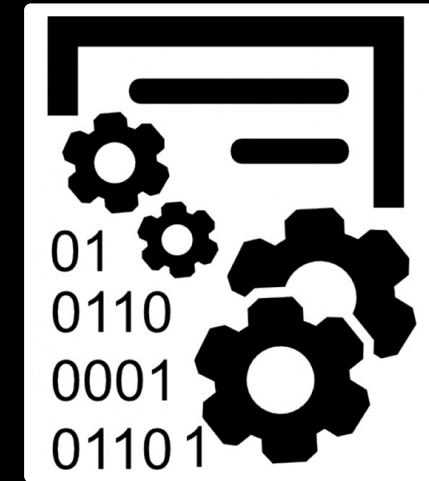
Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#4>

LOOPS

```
for (var i = 1; i < 100; i++) {  
  console.log(i);  
}
```

Повторения на блокове код

Работа на живо в клас (лаб)



Задачи с цикли

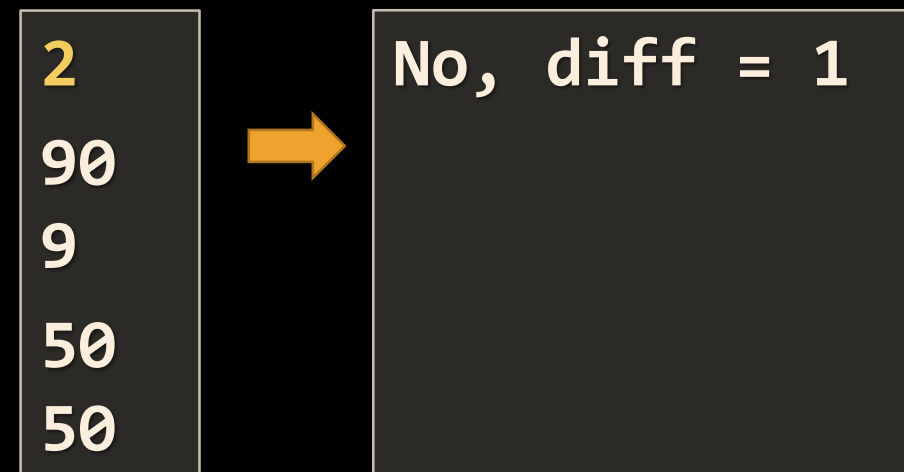
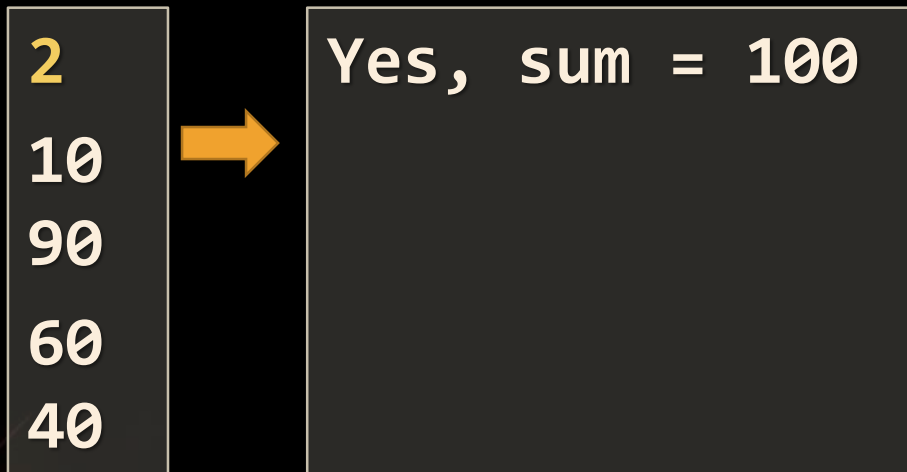
Техники за използване на for-цикли

Лява и дясна сума - условие

- Напишете програма, която:
 - Прочита цяло число(n) от потребителя
 - Прочита последователно $2*n$ -на брой числа
 - Проверява дали сумите на левите n -на брой и десните n -на брой числа са равни
 - При равенство извежда "Yes" и сумата, в противен случай - "No" и разликата (изчислена като положително число)

Лява и дясна сума – условие (2)

- Примерен вход и изход:



Лява и дясна сума - решение

```
function leftRightSum(args) {  
    let n = Number(args[0]);  
    let leftSum = 0;  
    for (let i = 1; i <= n; i++) {  
        leftSum = leftSum + Number(args[i]);  
    }  
    // TODO: read and calculate the rightSum  
    if (leftSum == rightSum) {  
        console.log("Yes, sum = " + leftSum);  
    } else {  
        // TODO: print the difference  
    }  
}
```

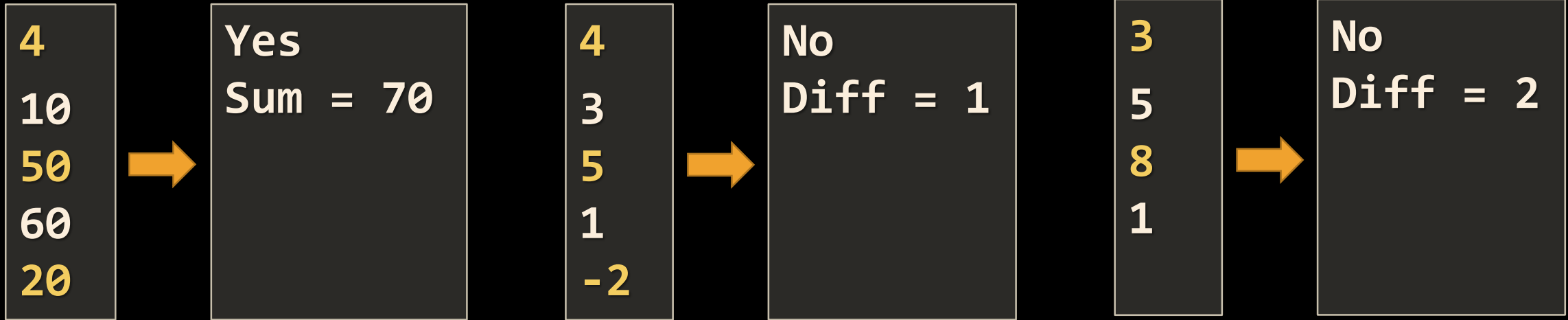
Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#6>

Четна / нечетна сума - условие

- Напишете програма, която:
 - Прочита цяло число(**n**) от потребителя
 - Прочита **последователно n-на брой** числа
 - Проверява дали сумата на числата на **четни позиции** е равна на сумата на числата на **нечетни позиции**
 - При равенство печата "**Yes**" + **сумата**; иначе печата "**No**" + **разликата** (положително число).

Четна / нечетна сума – условие (2)

- Примерен вход и изход:



Четна / нечетна сума - решение

```
function oddEvenSum(args) {  
  let n = Number(args[0]);  
  let oddSum = 0;  
  let evenSum = 0;  
  for (let i = 1; i <= n; i++) {  
    let element = Number(args[i]);  
    if (i % 2 !== 0) { oddSum += element; }  
    else { evenSum += element; }  
  }  
  
  // TODO: print the sum / difference  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#7>

Сумиране на гласните букви - условие

- Да се напише програма, която въвежда **текст** (стринг)
 - Отпечатва **сумата на гласните букви** според таблицата по-долу:

а	е	и	о	и
1	2	3	4	5

- Примери:

hello → **6** ($e+o = 2+4 = 6$)

hi → **3** ($i = 3$)

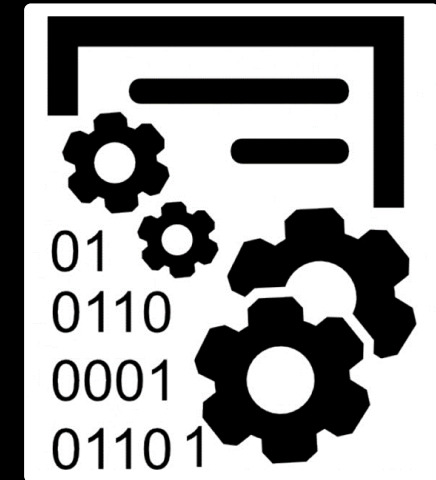
bamboo → **9** ($a+o+o = 1+4+4 = 9$)

beer → **4** ($e+e = 2+2 = 4$)

Сумиране на гласни букви - решение

```
function vowelSum([arg1]) {  
  let s = arg1;  
  let sum = 0;  
  for (var i = 0; i < s.length; i++) {  
    if (s[i] == 'a') { sum += 1; }  
    else if (s[i] == 'e') { sum += 2; }  
    // TODO: Check the other vowels  
  }  
  console.log("Vowels sum = " + sum);  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/Practice/Index/154#8>



Задачи с цикли

Работа на живо в клас (лаб)

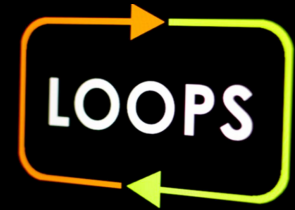
Какво научихме днес?

- Можем да **инкрементираме/декрементираме** числови стойности

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

- Можем да повтаряме блок код с **for**-цикъл:

```
for (let i = 1; i <= 10; i++) {  
  console.log("i = " + i);  
}
```



Какво научихме днес? (2)

- Можем да четем поредица от **n** числа:

```
function readInput(args) {  
  let n = Number(args[0]);  
  for (let i = 0; i < n; i++) {  
    let num = Number(args[i]);  
  }  
}
```



- Можем да вземем символ по индекс от текст

```
let text = "text";  
let symbol = text[2];  
console.log(symbol); // x
```



Повторения (цикли)



Въпроси?



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



Безплатни обучения в СофтУни

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

