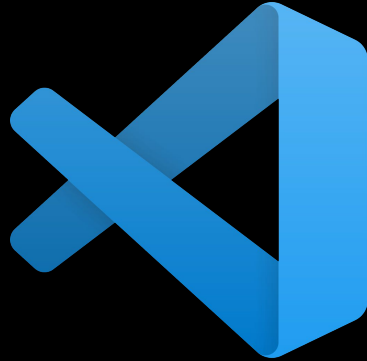




JavaScript

STUDIA PODYPLOMOWE
POLITECHNIKA BIAŁOSTOCKA

VS Code



<https://code.visualstudio.com/>

Node.js



<https://nodejs.org/en>

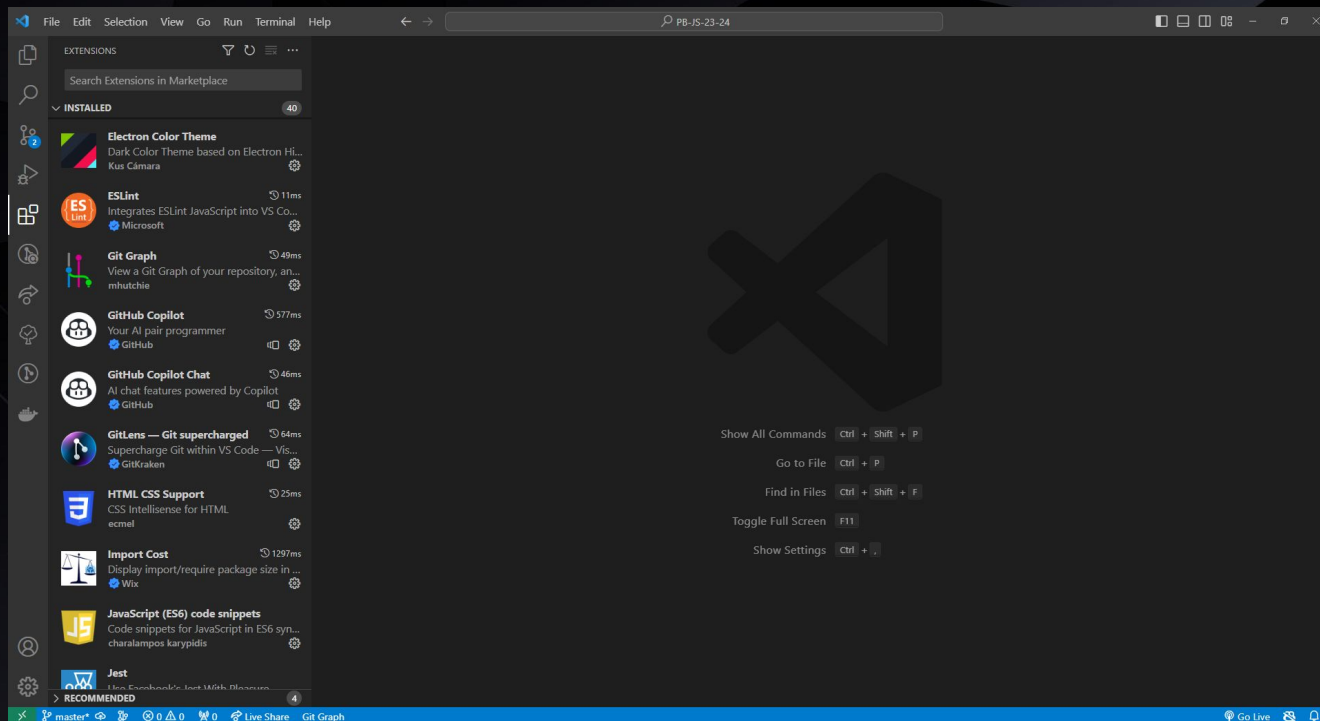
Browser



VS Code – colorized brackets

1. CTRL + Shift + P
2. Preferences: Open User Settings(JSON)
3. "editor.bracketPairColorization.enabled": true
4. CTRL + S

VS Code – plugins

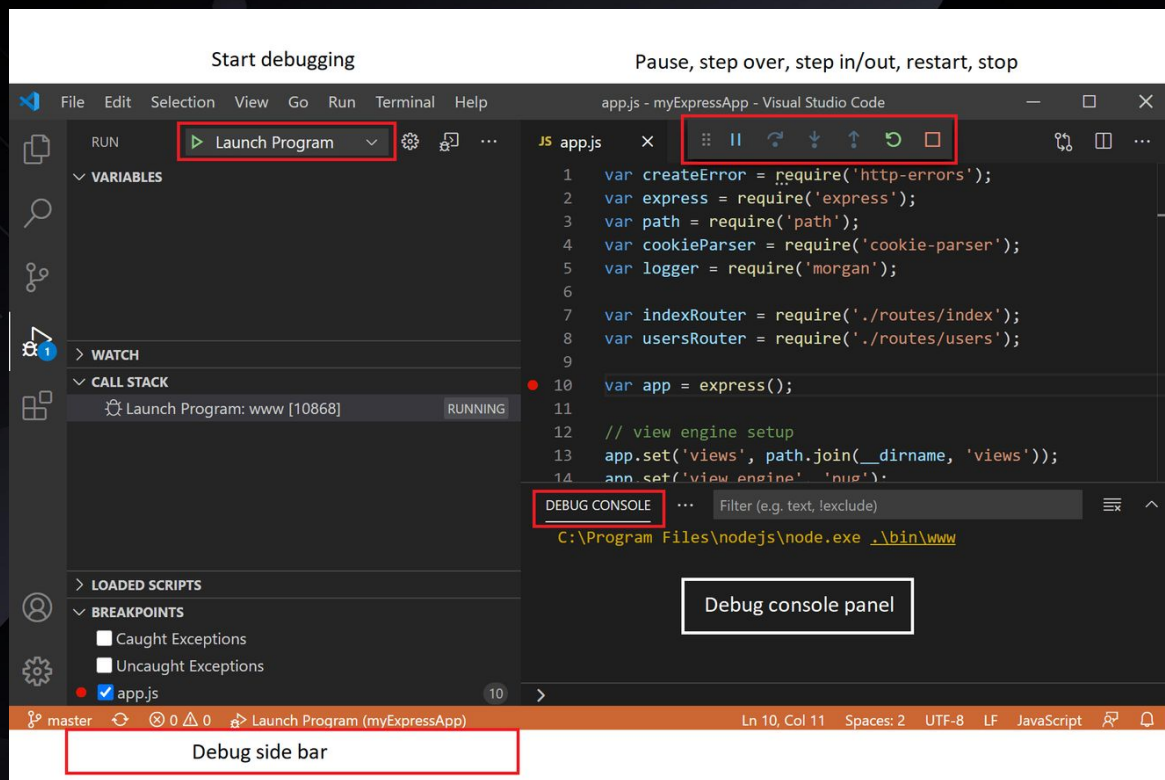


1. Prettier
2. ESLint
3. ...

VS Code – shortcuts

1. Comment – CTRL + /
2. Select same name – CTRL + D
3. Rename – F2
4. Copy line – ALT + SHIFT + ↓
5. Move line – ALT + ↓
6. Insert line below – CTRL + Enter

VS Code - debug



Types coercion



```
> ('b' + 'a' + + 'a' + 'a').toLowerCase();  
< 'banana'
```

```
> typeof NaN           > true==1  
< "number"             < true  
  
> 9999999999999999999 > true===1  
< 100000000000000000 < false  
  
> 0.5+0.1==0.6         > (!+[]+[]+![]).length  
< true                  < 9  
  
> 0.1+0.2==0.3         > 9+"1"  
< false                 < "91"  
  
> Math.max()           > 91-"1"  
< -Infinity            < 90  
  
> Math.min()           > []==0  
< Infinity              < true  
  
> []+[]                 > []+{}  
< ""                    < "[object Object]"  
  
> []+{}                 > {}+[]  
< ""                    < 0  
  
> true+true+true===3   > true  
< true                  < 0  
  
> true-true  
< 0
```



Types coercion

Automatic (**implicit**) conversion from one data type to another.

Type conversion can be forced (**explicit**)

<https://tc39.es/ecma262/#sec-abstract-operations>

Types coercion

Implicit coercion

```
1
2 0 == "0" // true, "0" is converted to number, Number(0)
3 0 == [] // true, [] is converted to number, Number([])
4 "0" == [] // false, [] is converted to string, String([])
5
```

String reverse

Write loop that will reverse given string.

What we need:

- input
- variable to store result
- iterations count
- condition to check against

Pen and Paper



Isolate the problem



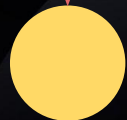
Example on small data



Find patterns



Write down algorithm



Translate to code

Mario

- Create Mario obstacle generator
- Based on given height number, it should generate half of a pyramid as shown below
- Output result to console



```
#  
##  
###  
####  
#####  
#####  
#####  
#####
```

```
1
2 // Define problem - space to # ratio
3
4 // display S S S #
5 // display S S # #
6 // display S # # #
7 // display # # # #
8
9 // Recognize pattern
10
11 // For each line display number of # equal to line number
12 // and number of spaces equal to total lines minus current line number
13
14 // 1. Repeat n times, where n is height
15 // i. count spaces
16 // ii. count #
17 // iii. display spaces and # according to count
18
19 // Assume our height is 8
20
```




mario.js

```
1
2 let height = 8;
3
4 for (let i = 1; i <= height; i++) {
5   let hashesCount = i;
6   let spacesCount = height - i;
7
8   let spaces = "";
9   for (let j = 0; j < spacesCount; j++) {
10     spaces += " ";
11   }
12
13   let hashes = "";
14   for (let k = 0; k < hashesCount; k++) {
15     hashes += "#";
16   }
17
18   console.log(spaces + hashes);
19 }
```


Display

- Create digital display with characters limit per line
- Input – string with unknown length
- Show as many words per line as possible without breaking them
- Break lines only on spacebar, skip break line spaces
- Output result to console
- Word in strings won't be longer than given limit

Output result

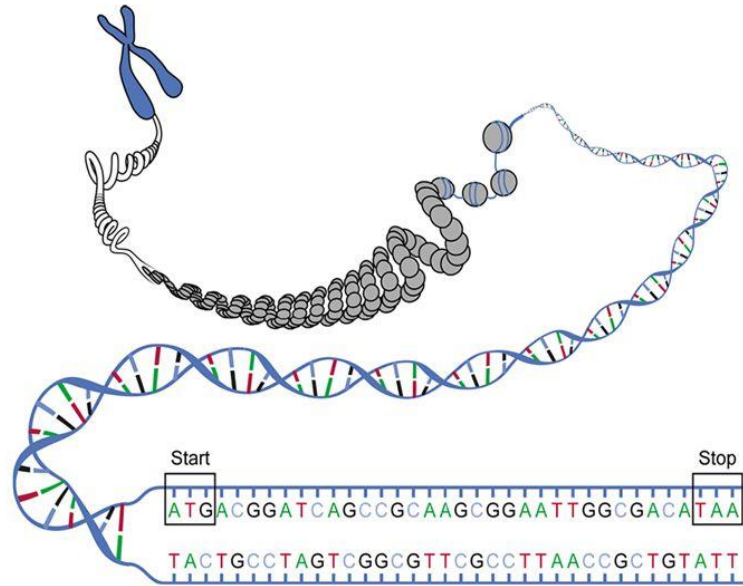
```
let screenSize = 22;  
let sentence = 'Hello, this is my long sentence, please break it up on the display.';  
"Hello, this is my long"  
"sentence, please break"  
"it up on the display."
```

sentence.js

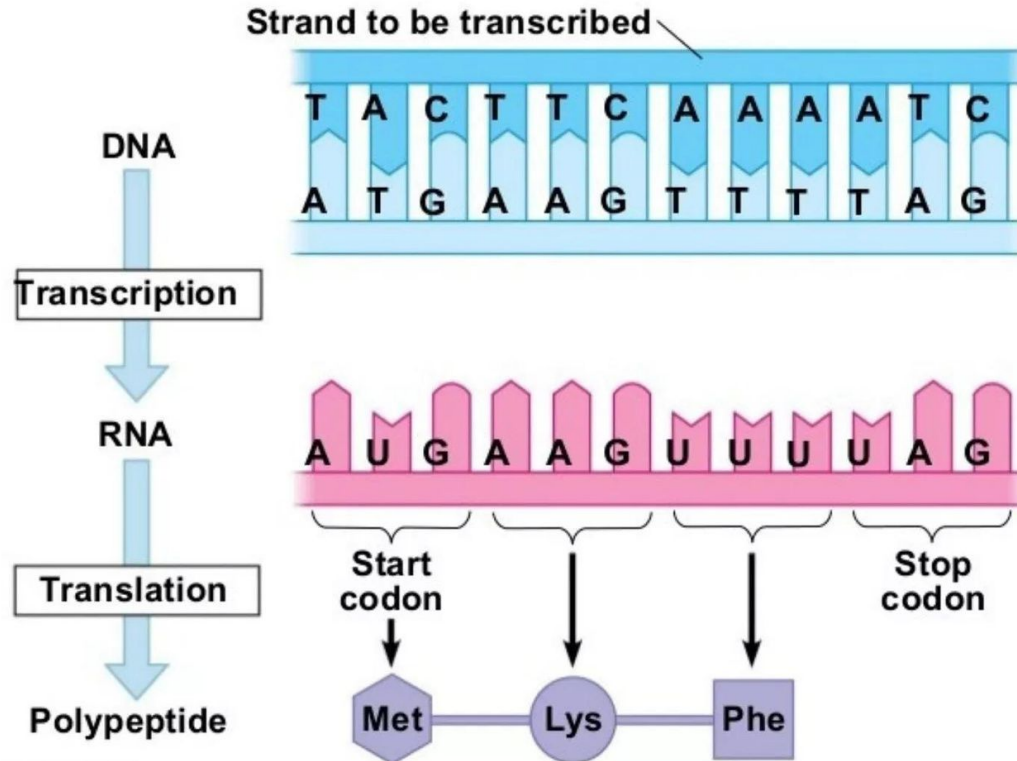
```
1
2 // 1. If sentence length < 12 - display
3 // 2. Else: As long as sentence is longer than 12 characters
4 // a. Take part of sentence that ends with space that is 13th character or lower
5 // b. Display it
6 // c. Save rest of sentence
7 // d. Repeat
8
```

DNA

DNA Structure



DNA



Gene finder

- Find a gene in a string of nucleotides
- Starting codon is - ATG
- Assume there is only one stop codon - TAA
- Remember that gene consist of multiple triplets of nucleotides

gen-finder.js

```
1
2 // 1. Find index of ATG - start codon
3 // Repeat until found gene
4 // a. get index of next gene
5 // b. if -1 there is no gene
6 // c. else check if string from ATG index to TAA index can be divided by 3
7 // d. if so gene is found
8 // e. else repeat
9 // 2. Display found gene or 'No gene' if there is none
10
```

HOMework

- String tools
- Mario - advanced
- Gene finder - advanced
- Credit card validator