



C PROGRAMMING

AKADEMIA ROZWOJU PFR

Presented by: Marta Plantykowski

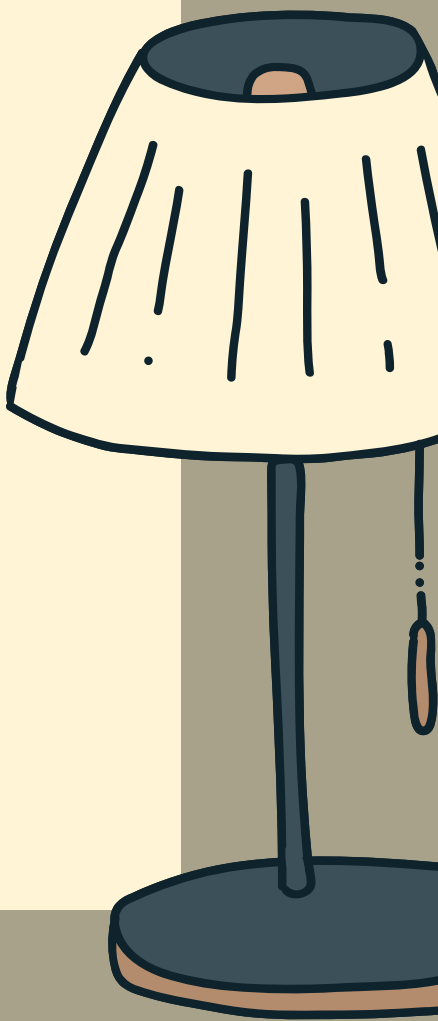
ZASADY

FORMA

- Otwarta rozmowa
- Komentarze czy wtrącenia są pożądane :)

PYTANIA

- Nie istnieją głupie pytania
- Pytania zadajemy po podniesieniu ręki lub na czacie



PROWADZĄCA

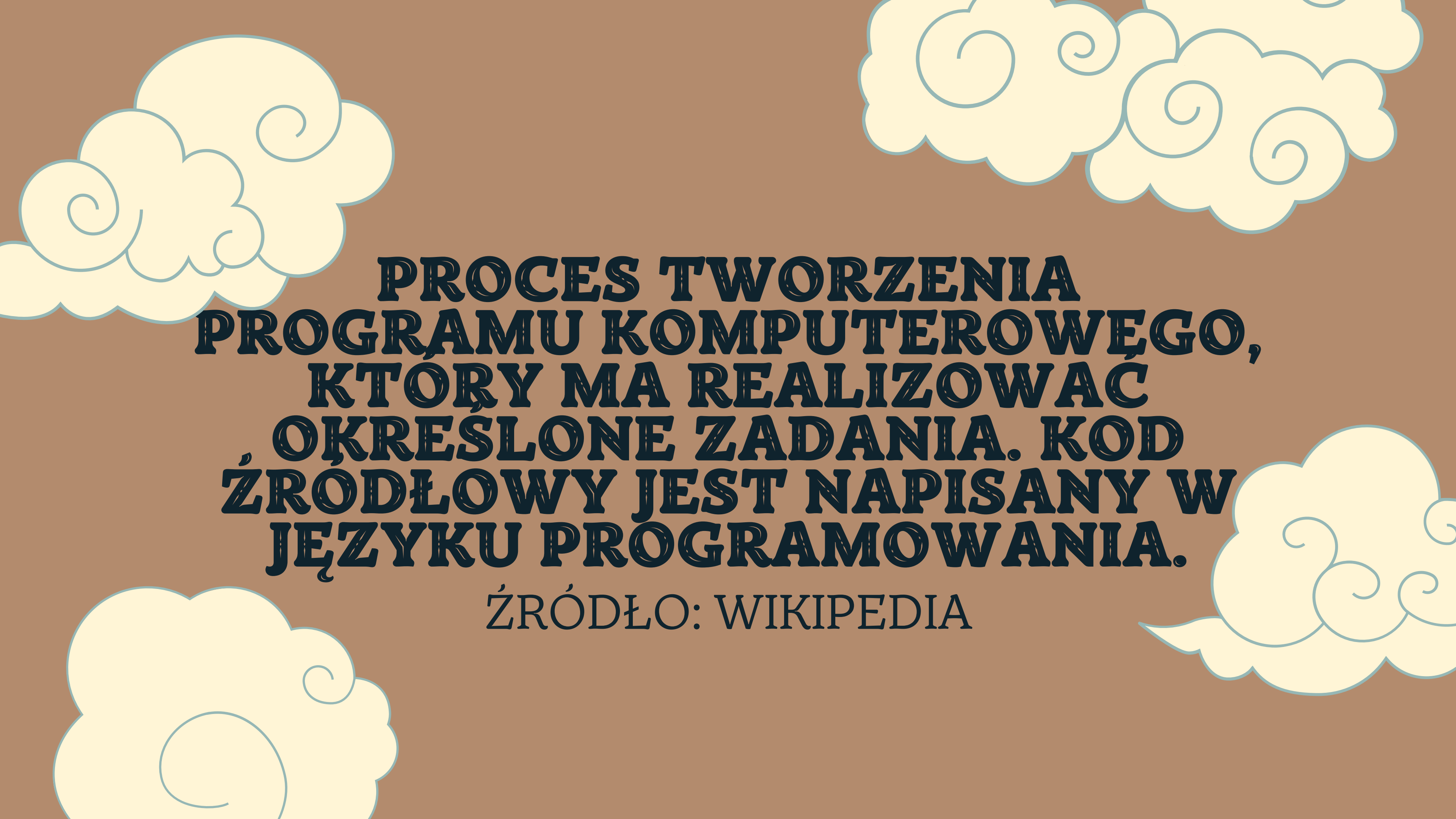


- Ukończyłam studia na kierunkach Automatyka i Robotyka oraz Informatyka na PG oraz studia podyplomowe Biznes.AI na ULK
- Pracuję zawodowo jako inżynier oprogramowania w firmie Intel Technology Poland (7 lat doświadczenia)
- Jestem członkinią Women In Artificial Intelligence (WAI) Poland
- Jestem CEO w projekcie HearAI

AGENDA



- Programowanie
- Środowisko (edytor, kompilator)
- Struktura programu
- Typy danych
- Zmienna
- Funkcja
- Operacje arytmetyczne i logiczne.

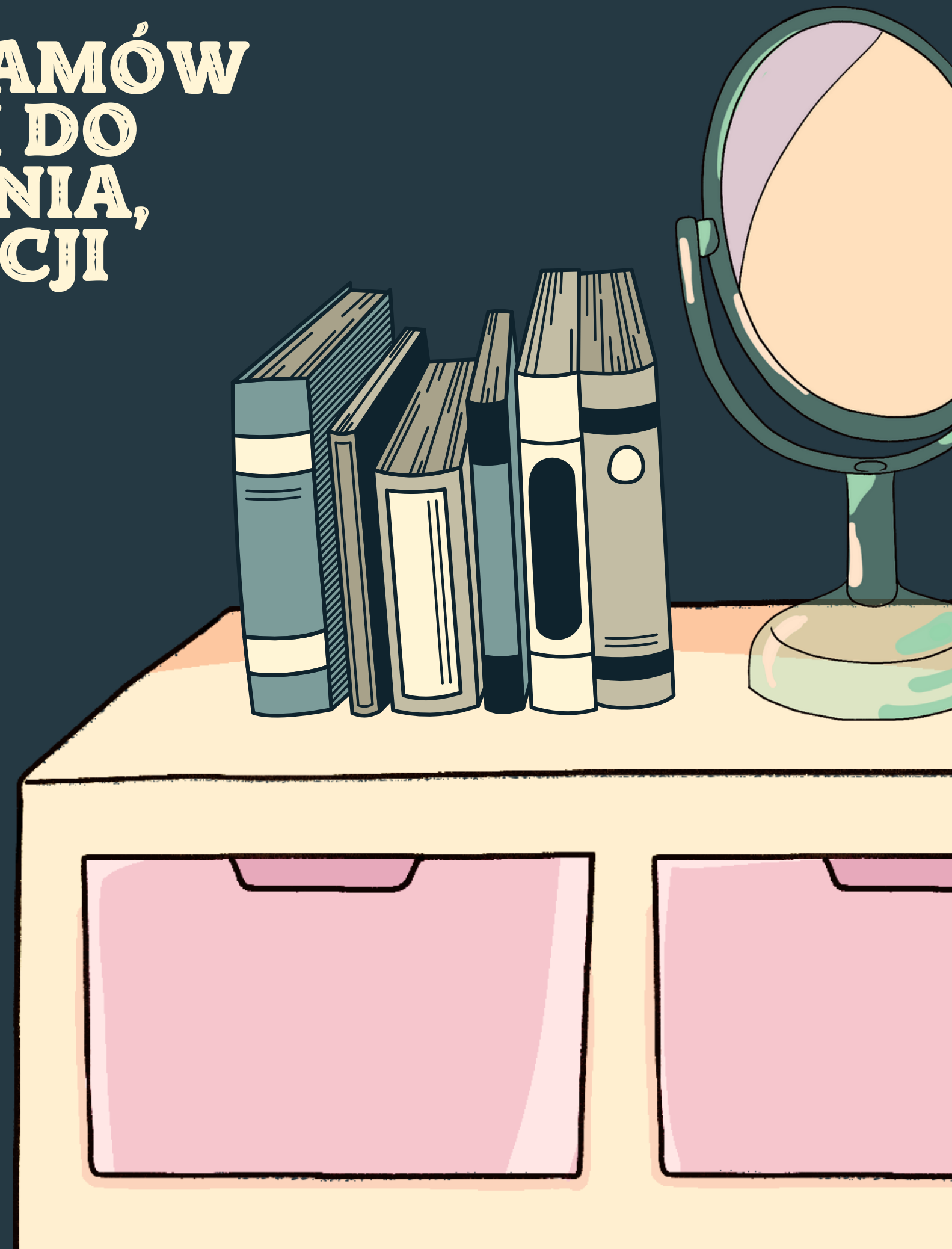


**PROCES TWORZENIA
PROGRAMU KOMPUTEROWEGO,
KTÓRY MA REALIZOWAĆ
OKREŚLONE ZADANIA. KOD
ŹRÓDŁOWY JEST NAPISANY W
JĘZYKU PROGRAMOWANIA.**

ŹRÓDŁO: WIKIPEDIA

**PROGRAM LUB ZESPÓŁ PROGRAMÓW
(ŚRODOWISKO) SŁUŻĄCYCH DO
TWORZENIA, MODYFIKOWANIA,
TESTOWANIA I KONSERWACJI
OPROGRAMOWANIA.**

ŹRÓDŁO: WIKIPEDIA



ŚRODOWISKO C

WOLNOŚĆ WYBORU

1. Microsoft Visual Studio
2. CLion
3. Eclipse
4. Code::Blocks
5. CodeLite
6. NetBeans
- 7....



ŚRODOWISKO C

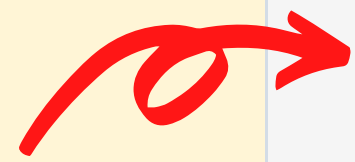
IN FACT..

1. Edytor kodu: Notatnik?
2. Kompilator

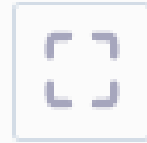
Na potrzeby szkolenia:

<https://www.programiz.com/c-programming/online-compiler/>





main.c

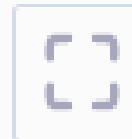


Run

Nazwa pliku

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

main.c



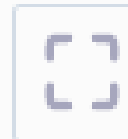
Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

Uruchamianie

main.c



Run

1 // Online C compiler to run C program online

2 #include <stdio.h>

3

4 int main() {

5 // Write C code here

6 printf("Hello world");

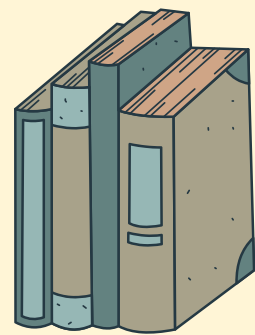
7

8 return 0;

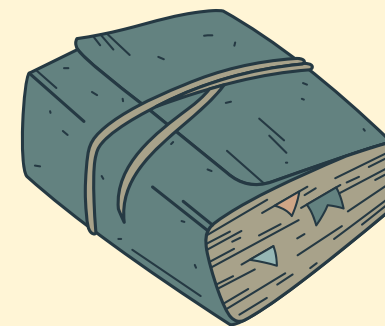
9 }

Numer linii
kodu

STRUKTURA PROGRAMU



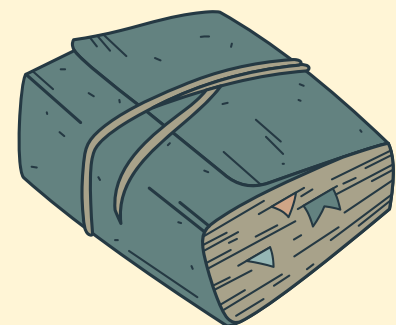
**KOMENDY PRE-
PROCESSORA**



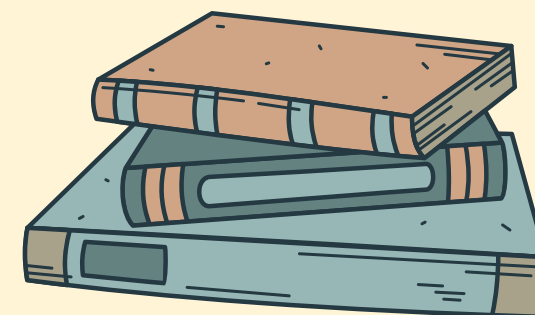
WYRAŻENIA



ZMIENNE

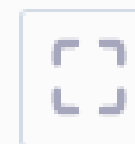


FUNKCJE



KOMENTARZE

main.c



Run

```
1 // Online C compiler to run C program online
```



Komentarz

```
2 #include <stdio.h>
```

```
3
```

```
4 int main() {
```



```
5     // Write C code here
```

Komentarz

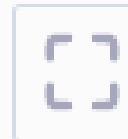
```
6     printf("Hello world");
```

```
7
```

```
8     return 0;
```

```
9 }
```

main.c



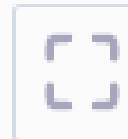
Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

Komeda
pre-procesora

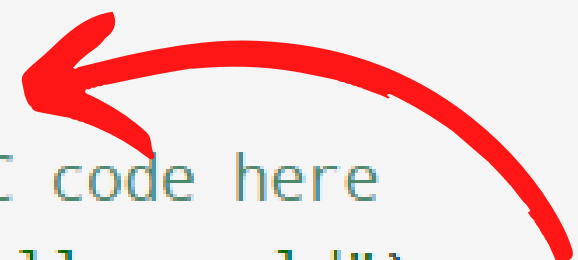


main.c



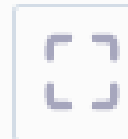
Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```



"Start"/
definicja funkcji

main.c

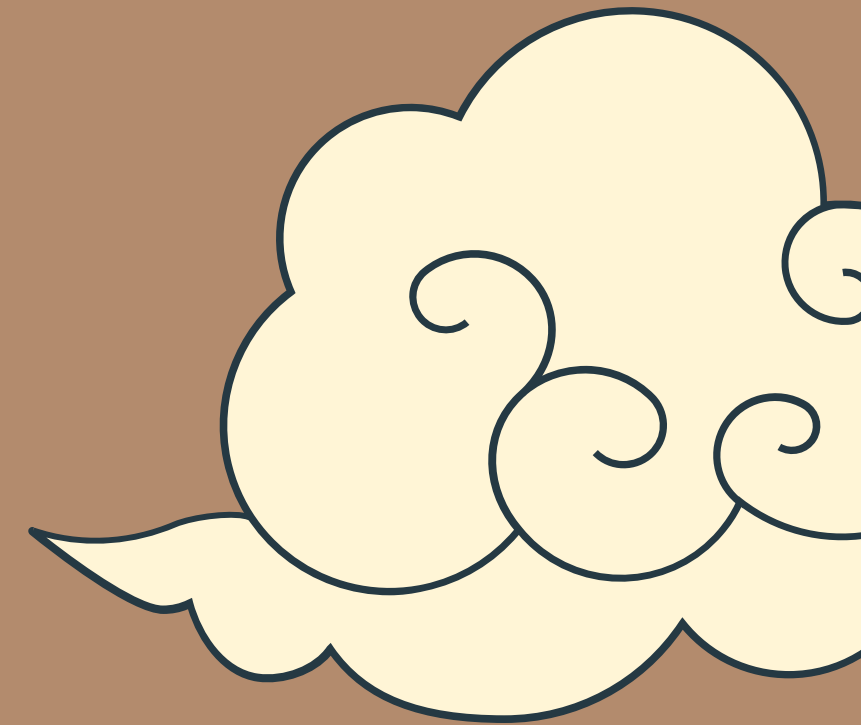


Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

Wywołanie
Funkcji

FUNKCJE



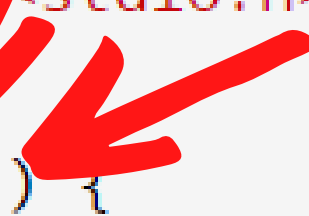
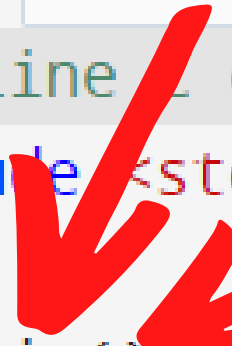
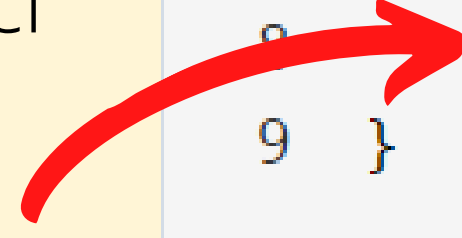
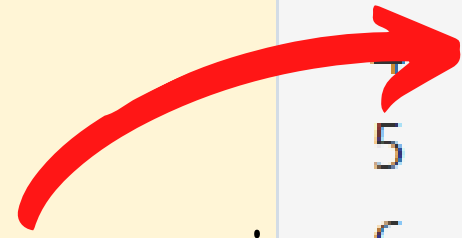
```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     // Write C code here
6     printf("Hello world");
7
8     return 0;
9 }
```

typ zwracanej
wartości

zwracana
wartość

parametry

ciało
funkcji



typ zwracanej
wartości

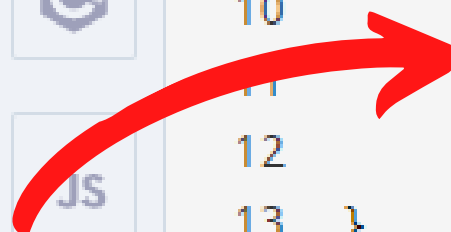
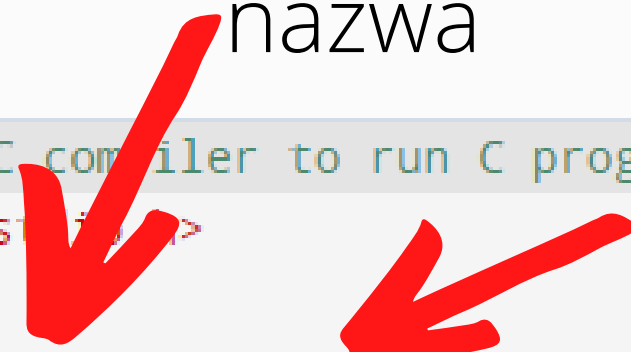
wywołanie
funkcji

```
main.c
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_Hello_world()
5 {
6     printf("Hello world");
7 }
8
9 int main() {
10    print_Hello_world();
11
12    return 0;
13 }
```

nazwa

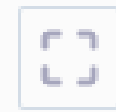
parametry

ciało
funkcji



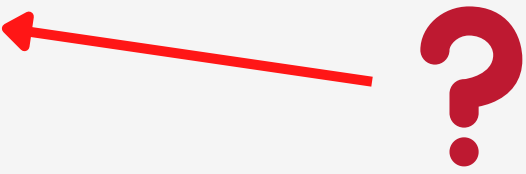


main.c



Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_Hello_world()
5 {
6     printf("Hello world");
7 }
8
9 int main() {
10     print_Hello_world();
11
12     return 0;
13 }
```



Komeda
pre-procesora

```
main.c [ ] [ ] [ Run ]  
1 // Online C compiler to run C program online  
2 #include <stdio.h>  
3  
4 int main() {  
5     // Write C code here  
6     printf("Hello world");  
7
```

Sr.No.	Functions & Description
1	printf() It is used to print the strings, integer, character etc on the output screen.
2	scanf() It reads the character, string, integer etc from the keyboard.
3	getc() It reads the character from the file.
4	putc() It writes the character to the file.
5	fopen() It opens the file and all file handling functions are defined in stdio.h header file.
6	fclose() It closes the opened file.
7	remove() It deletes the file.
8	fflush() It flushes the file.

Formatted Output and the printf function

One of the common task in every program is the printing of output. We use the output to request input from a user and later display the status/result, computations etc. In C programming there are several functions for printing formatted output. Here we discuss the printf() function, which writes output to the computer monitor. To use the printf() function we must include the stdio library in the source code. To do this just place the following code at the beginning of your program.

```
1 #include <stdio.h>
```

To print a simple message in computer screen you might call printf() function as follows:

```
1 #include <stdio.h>
2 main()
3 {
4 printf ("You are learning printf() function");
5 }
```

Output:

```
You are learning printf() function
```

In the above examples, the cursor will remain at the end of the printed output. If you repeat the above code in the following way the second message would appear immediately after the first one.

```
1 #include <stdio.h>
2 main()
3 {
4 printf("You are learning printf() function");
5 printf("You are learning printf() function");
6 }
```

Output:

```
You are learning printf() function You are learning printf() function
```

If we want to print the second output in a new line we must need a different way, which has discussed in the following section.



ZMIENNE



char
int

```
main.c Run  
1 // Online C compiler to run C program online  
2 #include <stdio.h>  
3  
4 int main() {  
5     char char1= "M";  
6     int int1 = 3;  
7  
8     printf("Hello world");  
9  
10    return 0;  
11 }
```


SIGNED

char	1B	[-128,127]
int	min 2B	[-32768; 32767]
	zwykle 4B	[-2147483648; 2147483647]

UNSIGNED

char	1B	[0, 255]
int	min 2B	[0, 65535]
	zwykle 4B	[0, 4294967295]

Źródło: http://fizyka.umk.pl/~tecumseh/EDU/PodstProgr1/PodstProgr1_I2.pdf



SIGNED

char	1B	[-128,127]
int	min 2B	[-32768; 32767]
	zwykle 4B	[-2147483648; 2147483647]

UNSIGNED

char	1B	[0, 255]
int	min 2B	[0, 65535]
	zwykle 4B	[0, 4294967295]

- Istnieją jeszcze: short i long
- Język C nie definiuje jednoznacznie wielkości typów podstawowych



char
int

```
main.c
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     char char1= "M";
6     int int1 = 3;
7
8     printf("Hello world");
9
10    return 0;
11 }
```



main.c



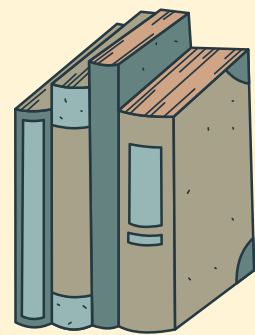
Run



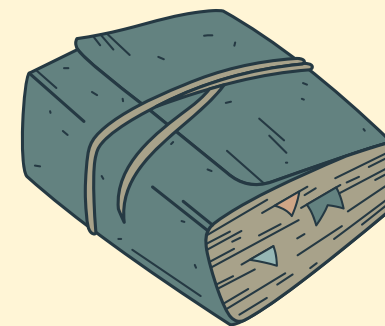
JS

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5     char char1= "M";
6     int int1 = 3;
7     int size;
8
9     size = sizeof(int);
10    printf("Size of int = %d\n", size);
11    size = sizeof(char);
12    printf("Size of char = %d\n", size);
13
14    return 0;
15 }
```

STRUKTURA PROGRAMU



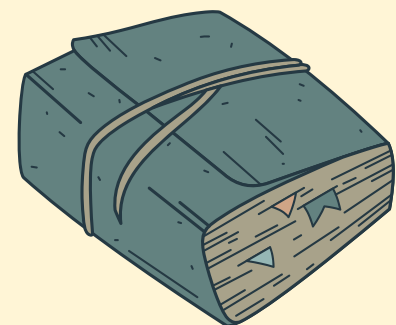
~~KOMENDY PRE-
PROCYSORA~~



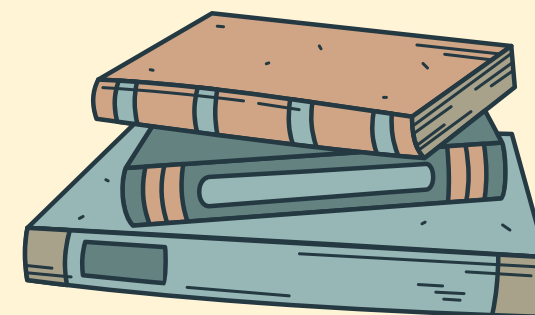
WYRAŻENIA



~~ZMIENNE~~

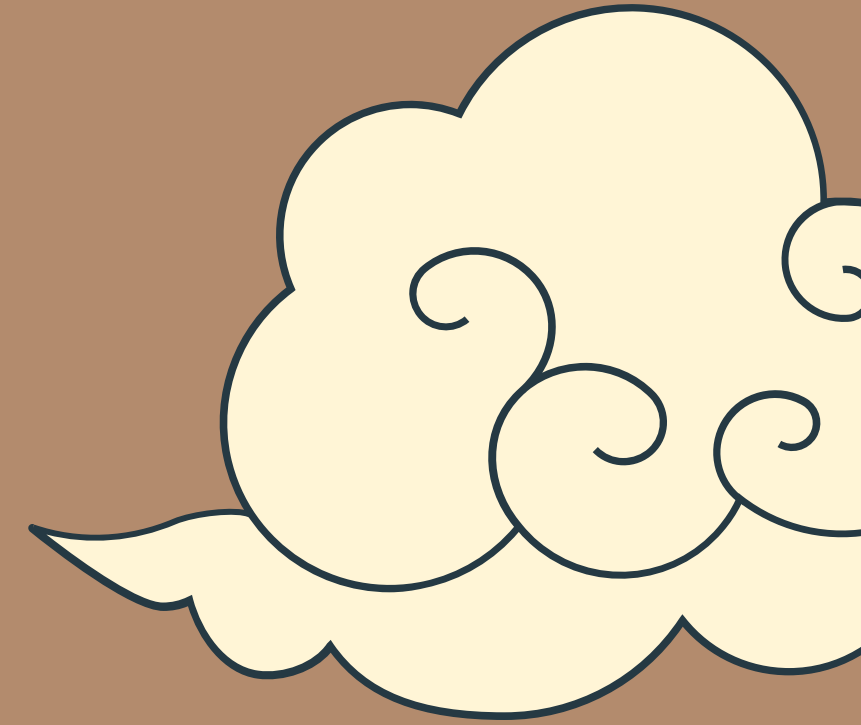
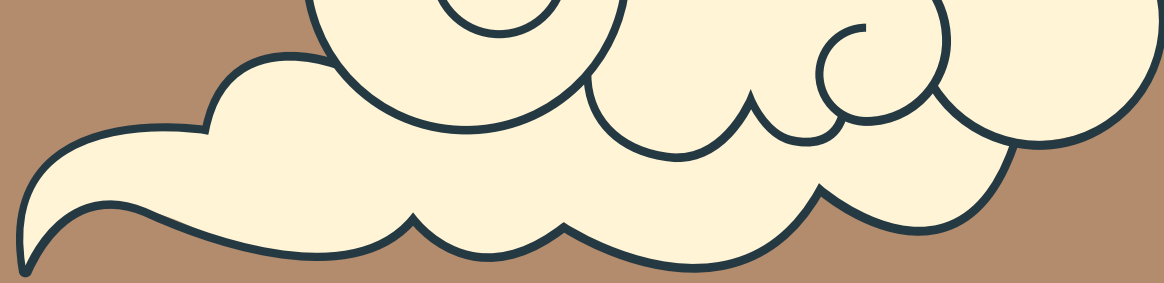


~~FUNKCJE~~



~~KOMENTARZE~~

WYRAŻENIA



= + - ; \ / * ...

O CZYM NALEŻY PAMIĘTAĆ?



ROZMIAR LITER MA ZNACZENIE!



KAŻDA KOMENDA KOŃCZY SIĘ ZNAKIEM ;



KOMPILATOR TO TWÓJ PRZYJACIEL

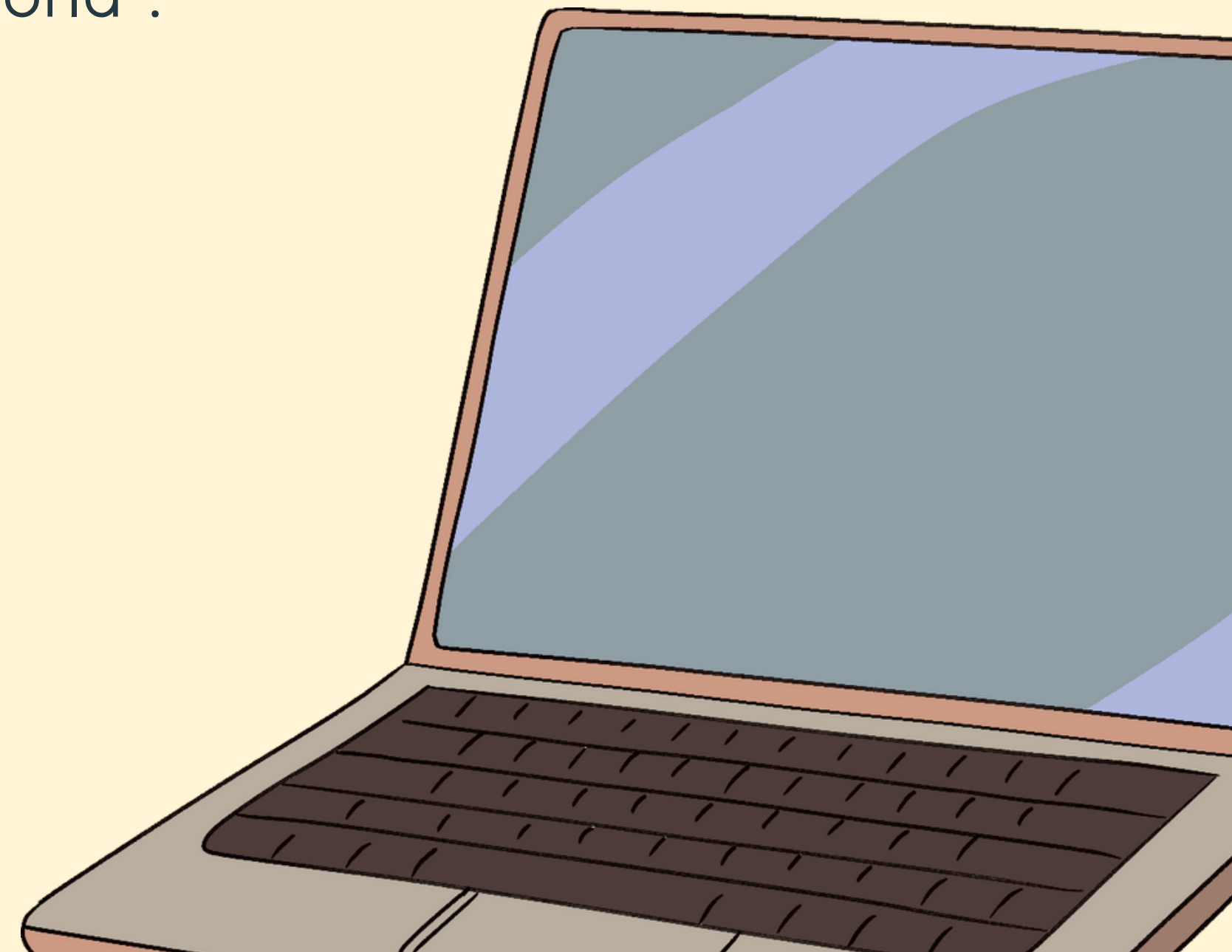
HANDS ON!

Zadanie

Napisz funkcję, która wydrukuje zwrot "Hello World".

Wywołaj ją w main.

Uruchom swój program i sprawdź, czy działa.





main.c



Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_Hello_world()
5 {
6     printf("Hello world");
7 }
8
9 int main() {
10     print_Hello_world();
11
12     return 0;
13 }
```

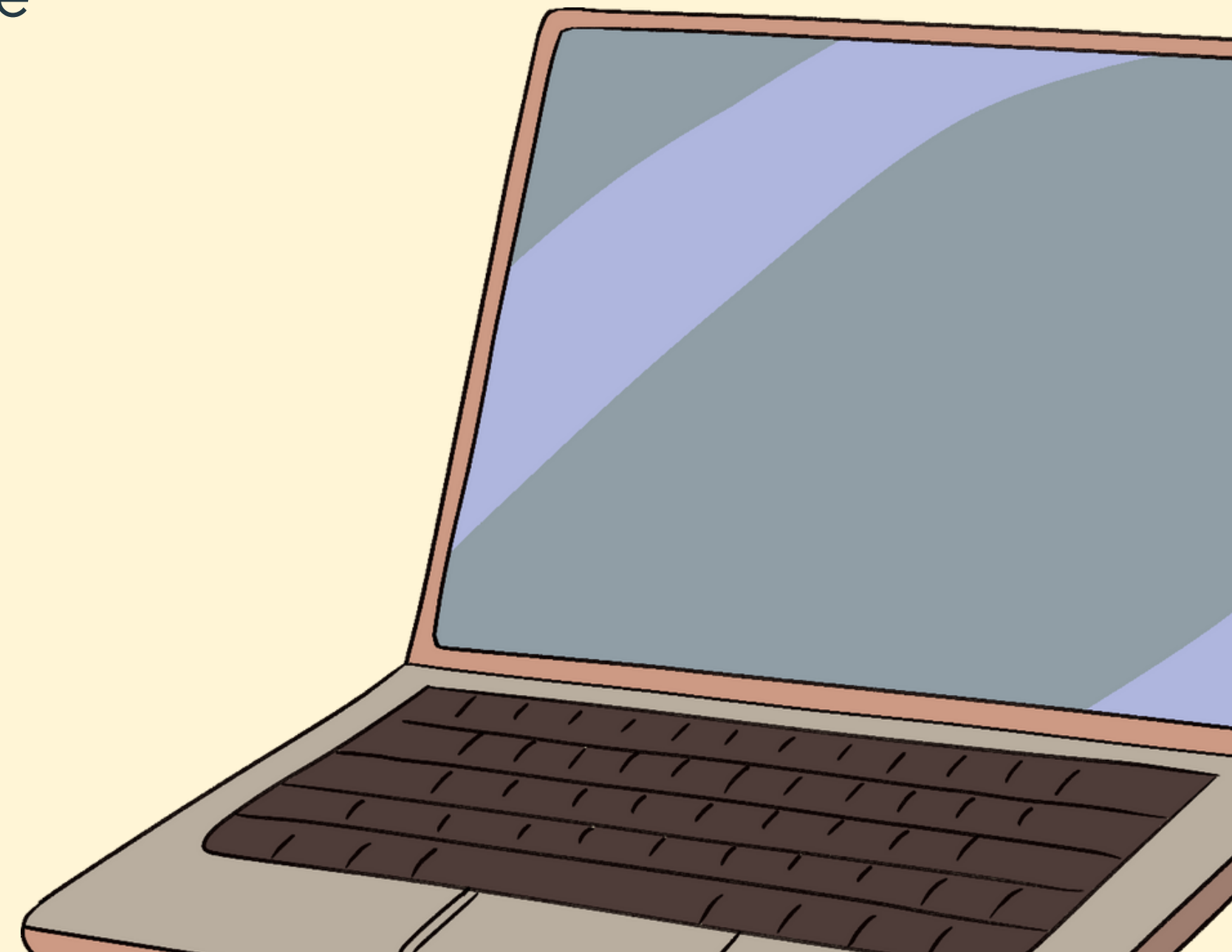
HANDS ON!

Zadanie

Napisz funkcję o nazwie `print_3`, która wydrukiuje numer "3".

Wywołaj ją w `main`.

Uruchom swój program i sprawdź, czy działa.





main.c



Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_3()
5 {
6     printf("3");
7 }
8
9 int main() {
10     print_3();
11     return 0;
12 }
```



JS

HANDS ON!

Zadanie

W main zdefiniuj zmienną typu int o nazwie val i zadeklaruj jej wartość równą 3.

W definicji funkcji print_3 dodaj parametr typu int o nazwie val.

Zmień wywołanie funkcji print_3 tak, aby uwzględniało nowy parametr.

Uruchom swój kod i sprawdź czy działa.

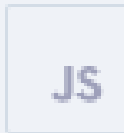




main.c



Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_3(int val)
5 {
6     printf("3");
7 }
8
9 int main() {
10     int val = 3;
11     print_3(val);
12     return 0;
13 }
```

%c	character		
%d	decimal (integer) number (base 10)		
%e	exponential floating-point number	\a	audible alert
%f	floating-point number	\b	backspace
%i	integer (base 10)	\f	form feed
%o	octal number (base 8)	\n	newline, or linefeed
%s	a string of characters	\r	carriage return
%u	unsigned decimal (integer) number	\t	tab
%x	number in hexadecimal (base 16)	\v	vertical tab
%%	print a percent sign		
\\%	print a percent sign		

The screenshot shows an online C compiler interface. On the left, there is a sidebar with icons for Python, C, JavaScript, and a JS button. The main editor area shows a C program named 'main.c' with the following code:

```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3 int main()
4 {
5     int testInteger = 5;
6     printf("Number = %d", testInteger);
7     return 0;
8 }

```

At the top right of the editor, there are icons for full screen, refresh, and a blue 'Run' button. To the right of the editor is an 'Output' window showing the result of the program execution:

```

/tmp/kqpZBsXQ4r.o
Number = 5

```

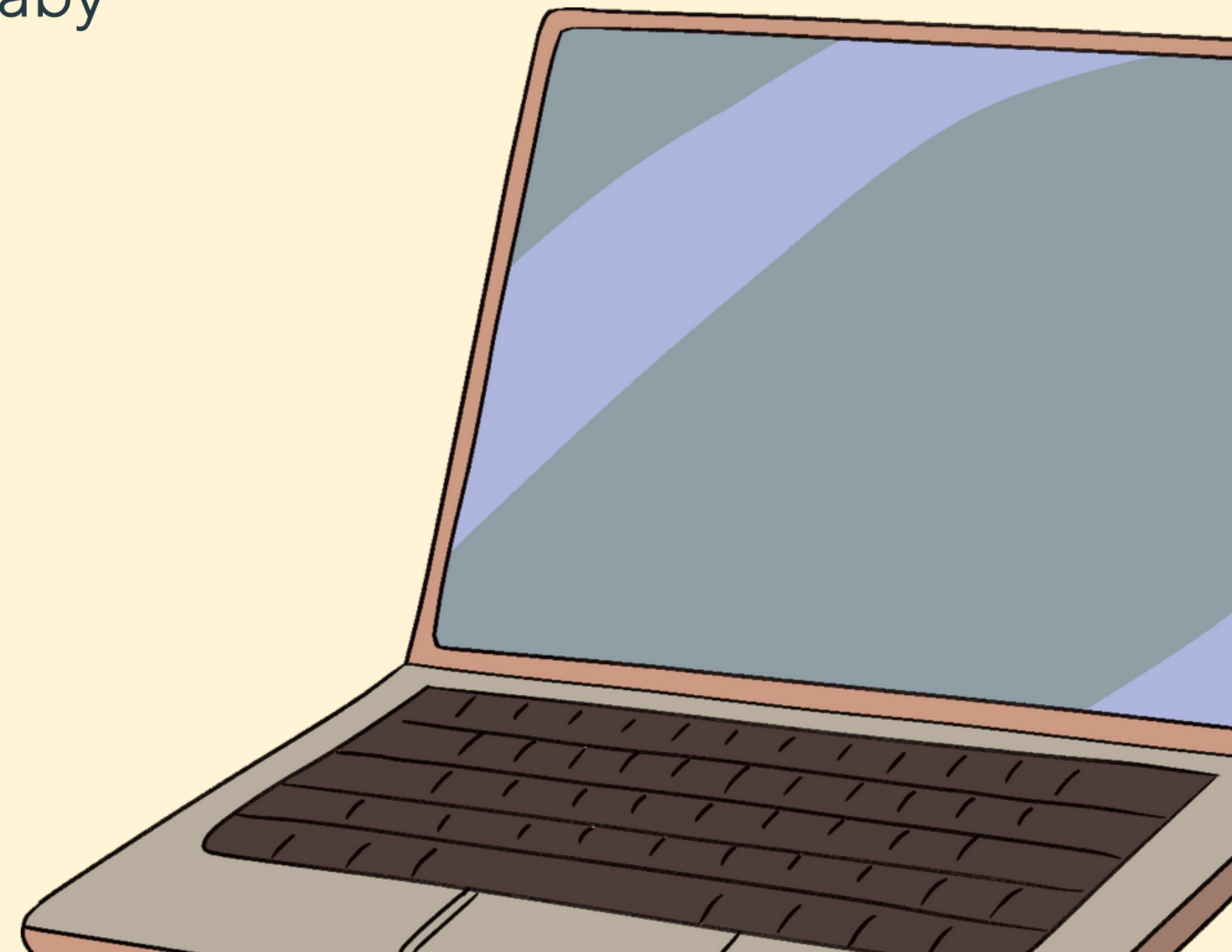
źródło: <https://alvinalexander.com/programming/printf-format-cheat-sheet/>

HANDS ON!

Zadanie

W definicji funkcji `print_3` zmodyfikuj `printf` tak, aby drukował wartość parametru.

Uruchom swój kod i sprawdź czy działa.





main.c



Run



JS

```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 void print_3(int val)
5 {
6     printf("%d\n", val);
7 }
8
9 int main() {
10     int val = 3;
11     print_3(val);
12     return 0;
13 }

```

%c	character
%d	decimal <u>integer</u> number (base 10)
%e	exponential floating-point number
%f	floating-point number
%i	integer (base 10)
%o	octal number (base 8)
%s	a string of characters
%u	unsigned decimal (integer) number
%x	number in hexadecimal (base 16)

\a	audible alert
\b	backspace
\f	form feed
\n	<u>newline</u> , or linefeed
\r	carriage return
\t	tab
\v	vertical tab
\\	backslash

HANDS ON!

Zadanie

Zdefiniuj nową funkcję o nazwie `add`.

Funkcja ma zwracać wartość typu `int` oraz przyjmować na wejście dwie zmienne typu `int` o nazwie `a` oraz `b`.

W ciele funkcji zadeklaruj zmienną typu `int` o nazwie `c` i przypisz jej wartość `0`.

Zwróć wartość `c` z funkcji.





main.c



Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int add(int a, int b)
5 {
6     int c = 0;
7
8     return c;
9 }
10
11 void print_3(int val)
12 {
13     printf("%d\n", val);
14 }
15
16 int main() {
17     int val = 3;
18     print_3(val);
19     return 0;
20 }
```



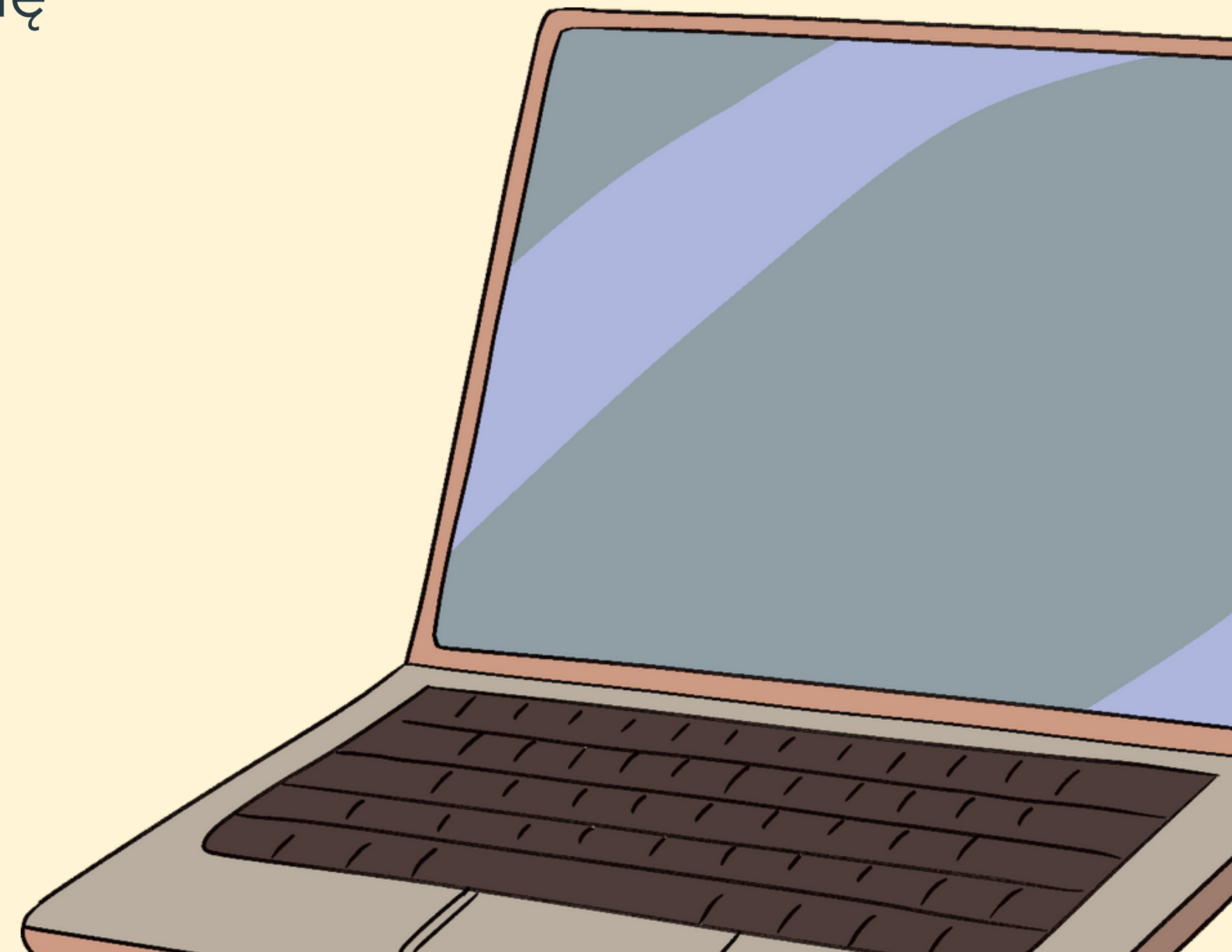
JS

HANDS ON!

Zadanie

W funkcji `add` ustaw wartość zmiennej `c` na sumę wartości `a` oraz `b`.

Uruchom swój program.





main.c



Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int add(int a, int b)
5 {
6     int c = a + b;
7
8     return c;
9 }
10
11 void print_3(int val)
12 {
13     printf("%d\n", val);
14 }
15
16 int main() {
17     int val = 3;
18     print_3(val);
19     return 0;
20 }
```

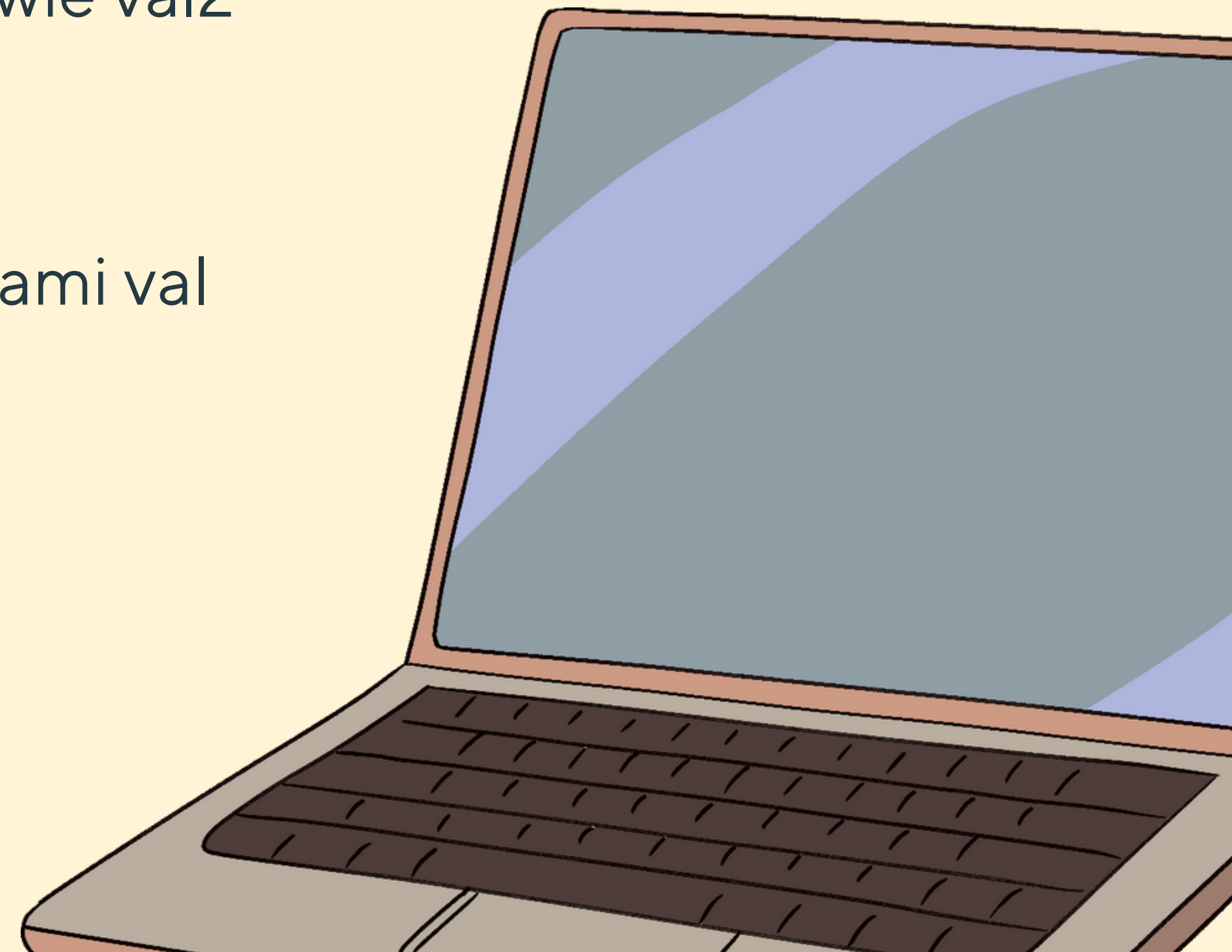
HANDS ON!

Zadanie

W funkcji main zdefiniuj zmienną typu int o nazwie val2 i przypisz jej wartość 5.

W funkcji main wywołaj funkcję add z parametrami val i val2.

Uruchom program.





main.c



Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int add(int a, int b)
5 {
6     int c = a + b;
7
8     return c;
9 }
10
11 void print_3(int val)
12 {
13     printf("%d\n", val);
14 }
15
16 int main() {
17     int val = 3;
18     int val2 =5;
19
20     add(val, val2);
21
22     print_3(val);
23     return 0;
24 }
```

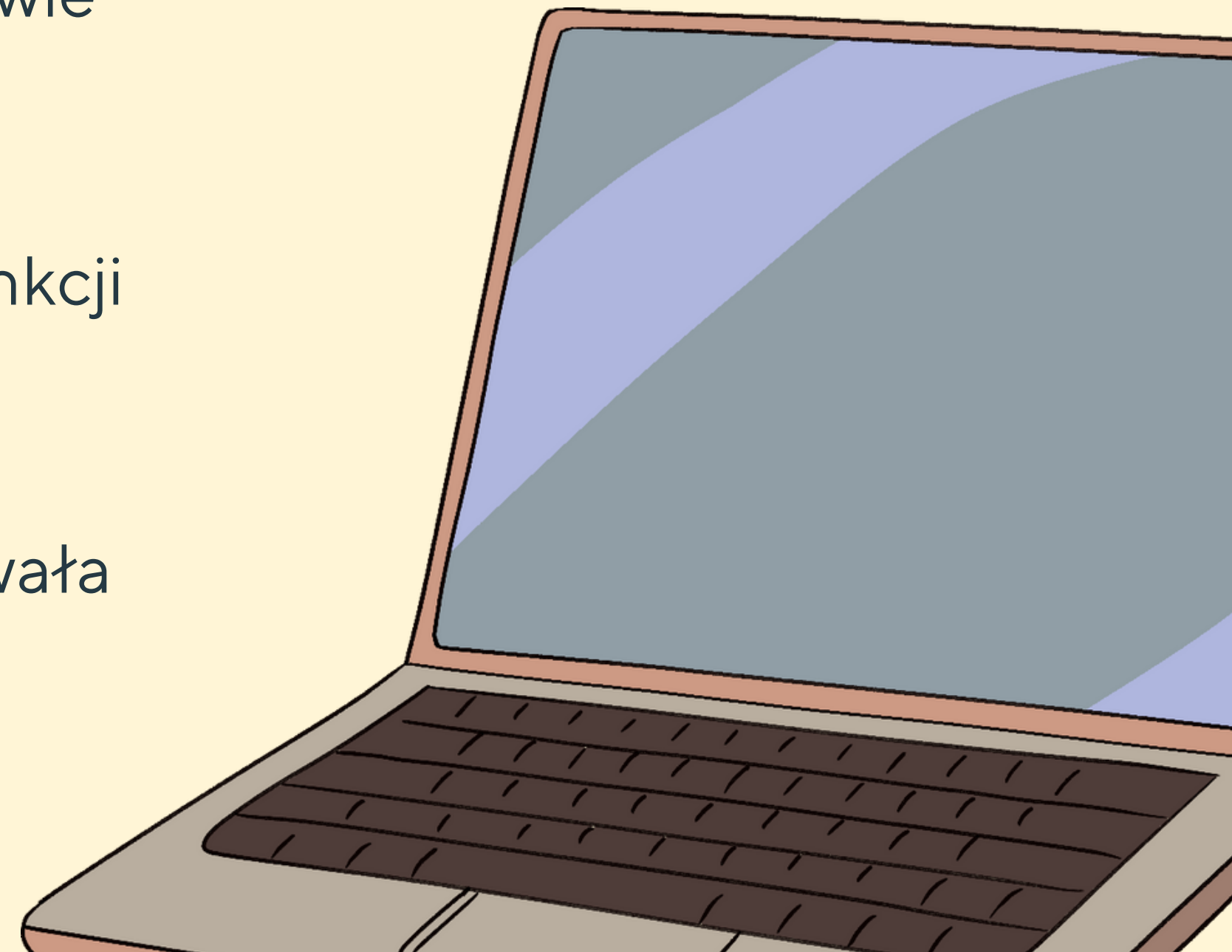
HANDS ON!

Zadanie

W funkcji main zdefiniuj zmienną typu int o nazwie result.

Przypisz zmiennej result wartość zwróconą z funkcji add.

Zmień wywołanie funkcji print_3 tak aby drukowała wartość zmiennej result.





main.c



Run



JS

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int add(int a, int b)
5 {
6     int c = a + b;
7
8     return c;
9 }
10
11 void print_3(int val)
12 {
13     printf("%d\n", val);
14 }
15
16 int main() {
17     int val = 3;
18     int val2 = 5;
19     int result;
20
21     result = add(val, val2);
22
23     print_3(result);
24     return 0;
25 }
```

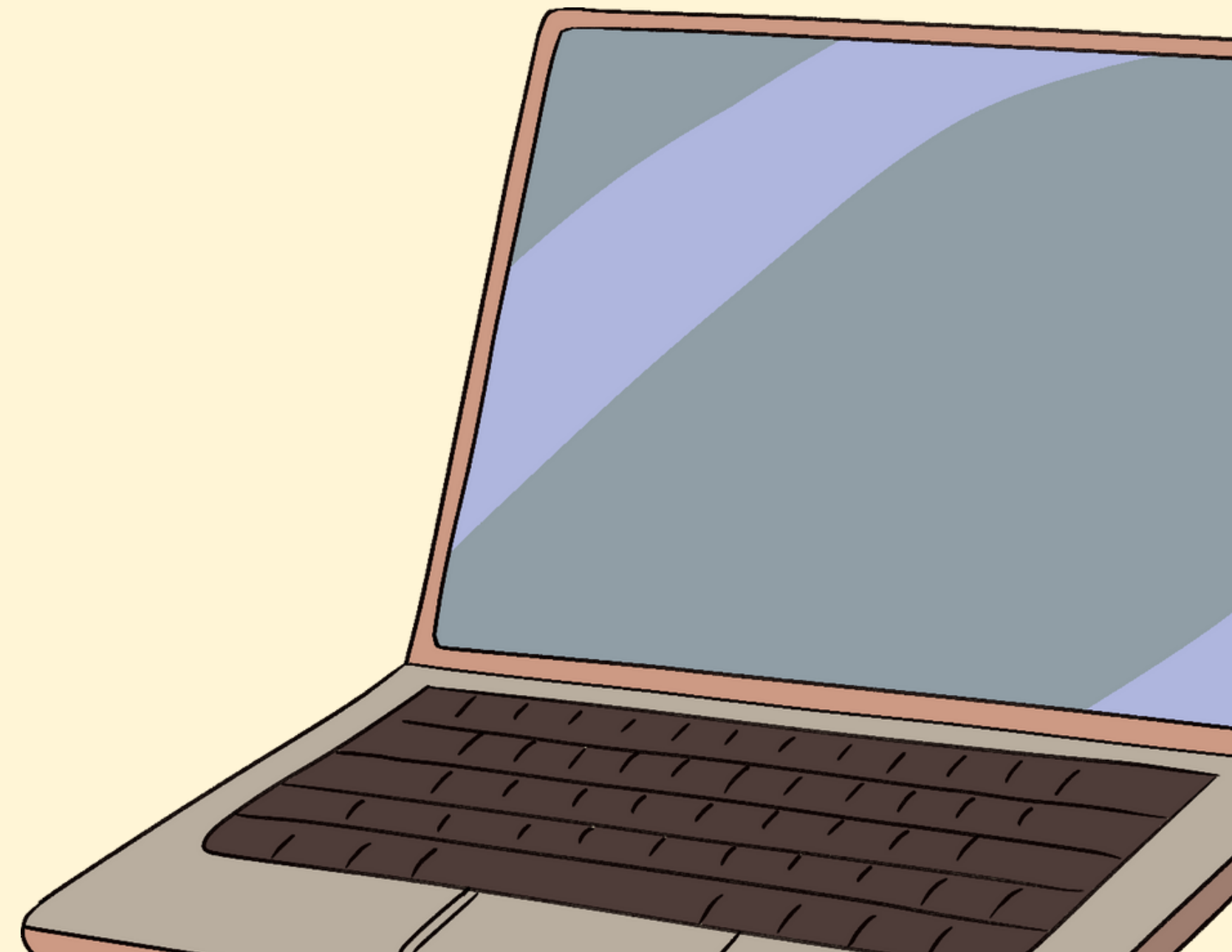
HANDS ON!

Zadanie

W wywołaniu funkcji `print_3` zmień parametr na wywołanie funkcji `add`.

Usuń zmienną `result`.

Uruchom swój program i sprawdź co się stanie.





main.c



Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int add(int a, int b)
5 {
6     int c = a + b;
7
8     return c;
9 }
10
11 void print_3(int val)
12 {
13     printf("%d\n", val);
14 }
15
16 int main() {
17     int val = 3;
18     int val2 = 5;
19
20     print_3(add(val, val2));
21     return 0;
22 }
```



JS

OPERATORY ARYTMETYCZNE

+ DODAWANIE

/ DZIELENIE

***** MOŻENIE

- ODEJMOWANIE

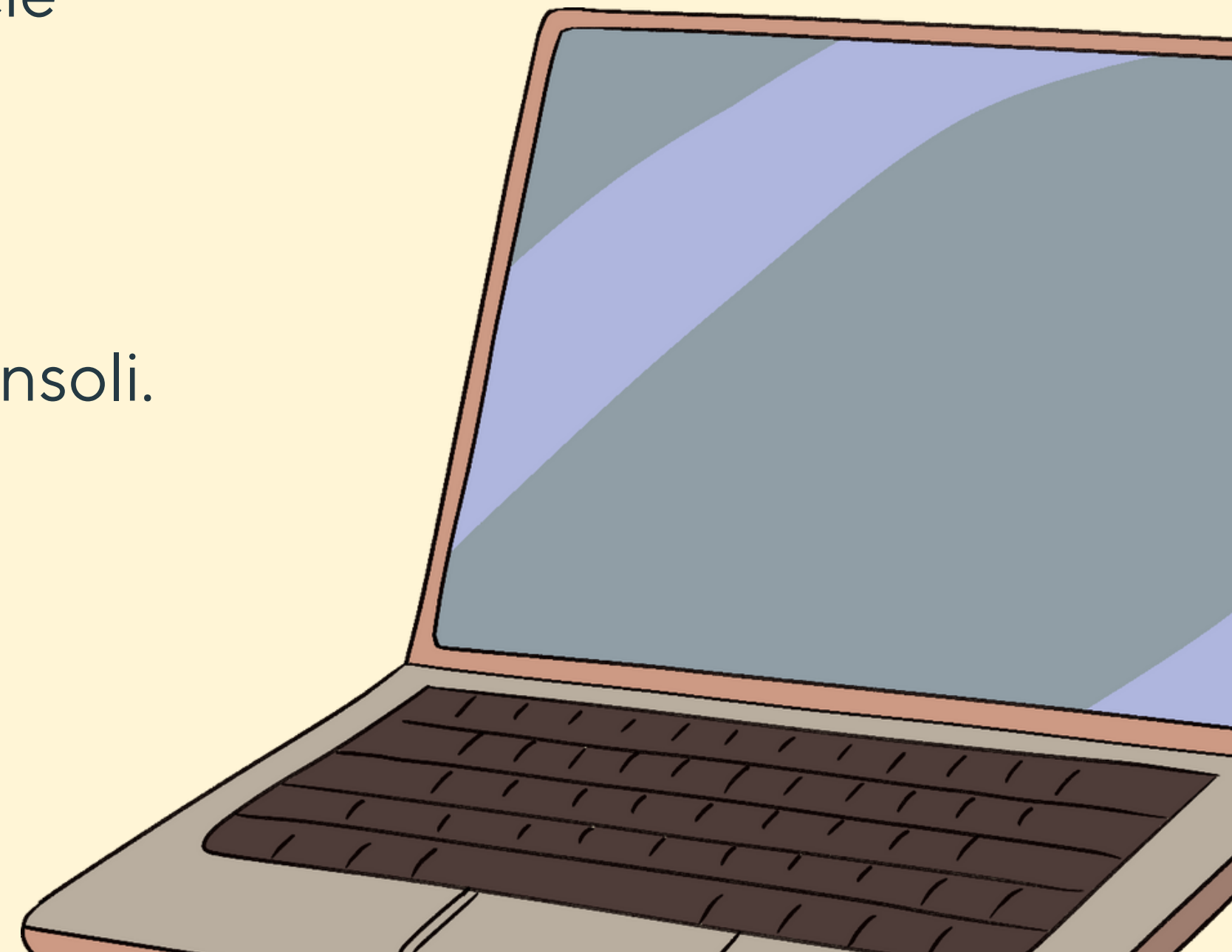
% MODULO

HANDS ON!

Zadanie

Napisz funkcję o nazwie `modulo`, która na wejście przyjmie dwie liczby całkowite i zwróci resztę z dzielenia.

Przetestuj swoją funkcję drukując wyście do konsoli.





main.c



Run

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int modulo(int a, int b)
5 {
6     int c = a % b;
7
8     return c;
9 }
10
11 int main() {
12     int val = 16;
13     int val2 = 5;
14
15     printf("%d\n", modulo(val, val2));
16     return 0;
17 }
```

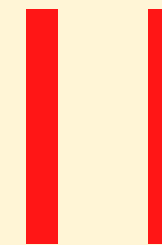


JS

OPERATORY LOGICZNE



ZAPRZECZENIE



ALTERNATYWA
(LUB)



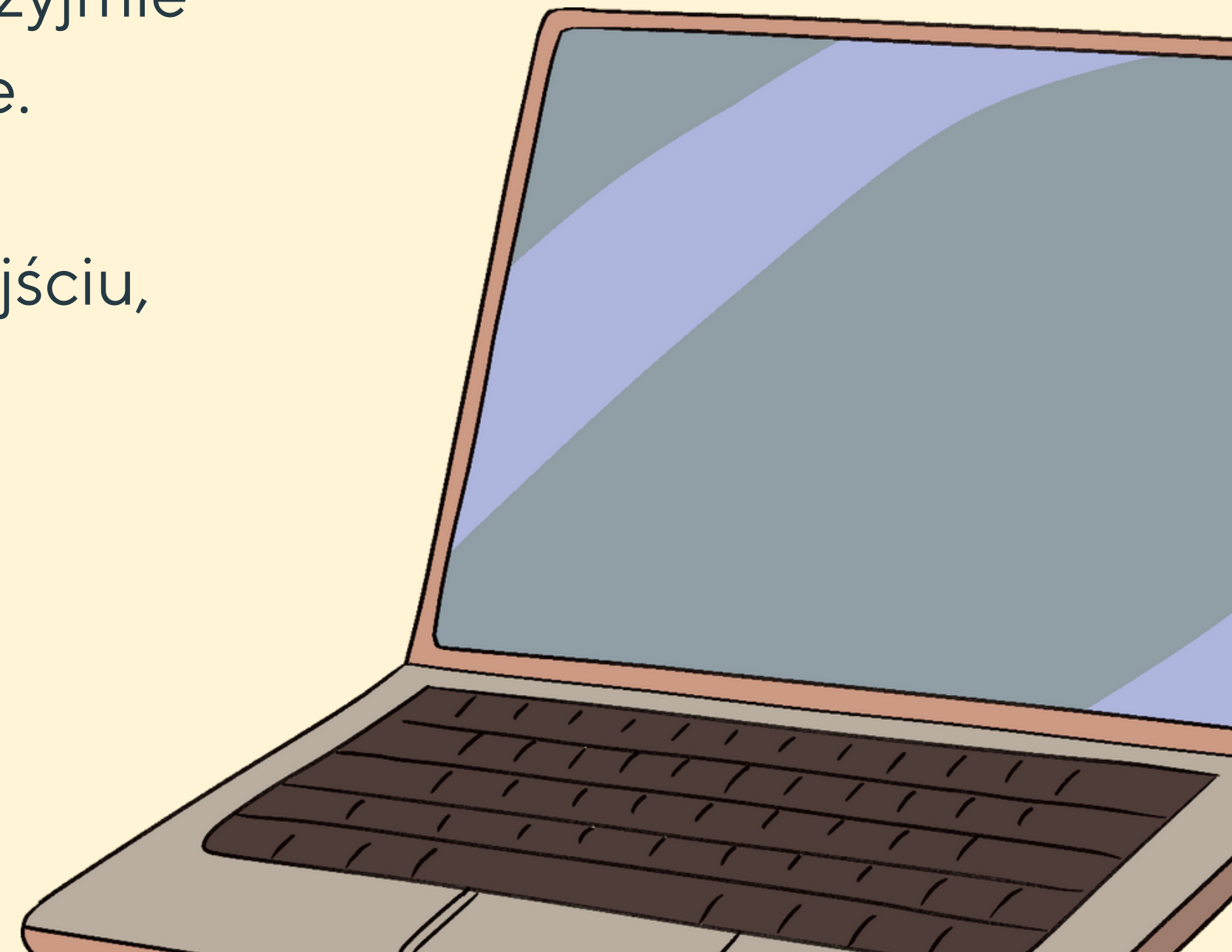
KONIUNKCJA (I)

HANDS ON!

Zadanie

Napisz funkcję o nazwie `nie`, która na wejście przyjmie jedną wartość typu `int` i zwróci jej zaprzeczenie.

Przetestuj swoją funkcję dla wartości 0 i 1 na wejściu, drukując je do konsoli.





main.c



Run



```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int nie(int a)
5 {
6     return !a;
7 }
8
9 int main() {
10     int val_false = 0;
11     int val_true = 1;
12
13     printf("%d\n", nie(val_false));
14     printf("%d\n", nie(val_true));
15     return 0;
16 }
```

PODSUMOWANIE



- Programowanie
- Środowisko
- Struktura programu
- Podstawowe typy danych
- Zmienna
- Funkcja
- Operacje arytmetyczne i logiczne.

CO DALEJ?

U
LinkedIn

YouTube

