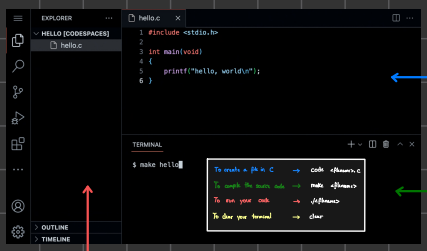
Lecture 1 - C

Hello World



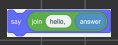
- text editor
  - you can edit your program using syntax highlighting
- command line interface (CLI)
  - you can send commands to the computer
- file explorer
  - you can find your files

Format Codes

Scratch → C

→ `printf "hello, world\n";`

→ `string answer = get_string "What's your name? ";`



→ `printf "hello, %s\n", answer;`

⚠ To print "100%" you should use two "%" signs to avoid errors.

Libraries and Documentation

What is `#include <stdio.h>`?

- it is a library that must be included at the beginning of your code
- it means Standard I/O library
- it contains the essential functions for C
- this course has its own library called cs50.h

Documentation

- it helps you to understand and get know the existing function in each library.
- for this course you can use a simplified version at manual.cs50.io

Conditionals

```c
if (x < y)
{
    printf("x is less than y\n");
}
else if (x > y)
{
    printf("x is greater than y\n");
}
else
{
    printf("x is equal to y\n");
}
```

```c
if (c == 'y' || c == 'Y')
{
    printf("Agreed.\n");
}
else if (c == 'n' || c == 'N')
{
    printf("Not agreed.\n");
}
```

(equal to, or)

⚠ For strings use " ", and for characters use ' '

Loops

```c
for (int i = 0; i < 3; i++)
{
    printf("meow\n");
}
```

(initialize variable; condition while TRUE; variable increment; repeated script)

```c
int n;
do
{
    n = get_init("Size: ");
}
while (n < 1);
```

(define variable; execute this script; if TRUE, repeat the 'do' script; if FALSE, stop repeating the script)

```c
counter = counter + 1;
counter += 1;
counter++;
```

TRUE = 1 or any other value
FALSE = 0

⚠ To break a loop press 'control + C'

⚠ Loops can be nested

Data Types

- bool → can be TRUE or FALSE
- char → holds one character (letter, number, etc)
- double → can represent fractional as well as whole values
- float → stores floating-point number values
- int → contains only whole numbers
- long → stores integers with more bits, counts higher than int
- string → stores values that are treated as text
- ...

Comments

- fundamental parts of a computer program, where you leave explanatory remarks to yourself and others that may be collaborating with you regarding your code.
- Comments involves placing // into your code, followed by a comment.

```c
int main(void)
{
    // Get size of grid
    int n;
    do
    {
        n = get_int("Size: ");
    }
    while (n < 1);
```

This is a comment

Operators

- + → for addition
- − → for substraction
- * → for multiplication
- / → for division
- % → for remainder

Abstraction

- is the art of simplifying our code such that it deals with smaller problems.
- looking at your code, you can see how two essential problems in our code are get size of grid and print grid of bricks
- we can abstract away these two problems into separate functions.
- we have the main function that calls two other functions called get_size and print_grid

```c
#include <cs50.h>
#include <stdio.h>

int get_size(void);
void print_grid(int n);

int main(void)
{
    int n = get_size();
    print_grid(n);
}

int get_size(void)
{
    int n;
    do
    {
        n = get_int("Size: ");
    }
    while (n < 1);
    return n;
}

void print_grid(int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("#");
        }
        printf("\n");
    }
}
```

Command Line Interface

- ls → lists files in a directory
- cd → changes our current directory
- cp → copies files and directories
- mkdir → makes a directory
- mv → moving (renaming) files and directories
- rm → removes (deletes) files
- rmdir → removes (deletes) directories