



Microcontrollers & Applications

Lecture 2.1: Comments & Variables & Constants

Comments (1)

- In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program. They are added for humans and are generally ignored by compilers and interpreters.
- C:

```
// single-line comment
```

```
a_command; // another short comment here
```

```
/*
```

You can write multiple-line comments.

```
@2024 by Yalcin Isler
```

```
*/
```

Comments (2)

- Python:

```
# single-line comment
```

```
a_command; # another short comment here
```

```
"""
```

```
You can write multiple-line comments.
```

```
@2024 by Yalcin Isler
```

```
"""
```

Variable Definition

- C: *Variables must be defined before their first use!*

```
variable_type variable_name;
```

```
variable_type variable_name = initial_value;
```

- Python: *Variables can be used directly!*

```
variable_name = its_value;
```

Variable Names

- A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume)
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the programming-language specific keywords

Variable Name Examples (Correct & False)

- myName
- my_name
- x
- i
- iTemperature
- bUserResponse
- x1
- x_0
- break
- if
- 0degrees
- 0_1_2
- name-surname
- final?or_continue?
- x\$
- x,y,z

Common Variable Types (C & Python)

- int:
 - stores integers (whole numbers), without decimals, such as 123 or -123
- float, double:
 - stores floating point numbers, with decimals, such as 19.99 or -19.99
- char:
 - stores single characters, such as 'a' or 'B'. Characters are surrounded by single quotes
- string:
 - stores multiple characters. Characters are surrounded by double quotes
 - `char greetings[] = "Hello World!";`

More Variable Types (C)

- Boolean Type: `boolean`
- Precision Specifiers: `long`, `short`
- Sign Specifiers : `signed`, `unsigned`
- `byte` = unsigned char
- `uint` = unsigned integer
- `struct` = structural variable type definition
- `union` = combining multiple variable type definition
- `enum` = enumerated (pre-defined) values

Struct and Union Types (C)

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

```
struct Point{
```

```
    int x;
```

```
    int y;
```

```
};
```

```
int main() {
```

```
    struct Point p1;
```

```
    p1.x = 1;
```

```
    p1.y = 3;
```

```
    printf("%d \n", p1.x);
```

```
    printf("%d \n", p1.y);
```

```
    return 0;
```

```
}
```

```
1
3
```

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

```
struct P2D{
```

```
    unsigned char x;
```

```
    unsigned char y;
```

```
};
```

```
union Point{
```

```
    unsigned int xy;
```

```
    struct P2D p2;
```

```
};
```

```
int main() {
```

```
    union Point p1;
```

```
    p1.xy = 256 * 3 + 1;
```

```
    printf("xy = %d \n", p1.xy);
```

```
    printf("x = %d \n", p1.p2.x);
```

```
    printf("y = %d \n", p1.p2.y);
```

```
    return 0;
```

```
}
```

```
xy = 769
x = 1
y = 3
```

Enum Type (C)

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

```
enum Level {  
    LOW,  
    MEDIUM,  
    HIGH  
};
```

```
int main() {  
    enum Level myVar = MEDIUM;  
    printf("%d", myVar);
```

```
    return 0;  
}
```

1

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

```
enum Level {  
    LOW = 25,  
    MEDIUM = 50,  
    HIGH = 75  
};
```

```
int main() {  
    enum Level myVar = MEDIUM;  
    printf("%d", myVar);
```

```
    return 0;
```

```
}
```

50

It is an ordered and numbered constant definition method 😊

```
#define LOW      0  
#define MEDIUM  1  
#define HIGH     2
```

or

```
#define LOW      25  
#define MEDIUM  50  
#define HIGH     75
```

More Variable Types (Python)

- Numeric Types: `complex`
- Sequence Types: `list, tuple, range`
- Mapping Type: `dictionary`
- Set Types: `set, frozenset`
- Boolean Type: `bool`
- Binary Types: `bytes, bytearray, memoryview`
- None Type: `NoneType`

Constants (C & Python)

- Starts with const keyword
- Similar to variables where their initial values cannot be changed in runtime
- All capital letters, not lowercase letters, in general
- `const int BIRTHYEAR = 1980;`
- `const char SCHOOL = "Izmir Katip Celebi University";`
- `#define BIRTHYEAR 1980`
- `#define SCHOOL "Izmir Katip Celebi University"`
- `#define` is similar to Find&Replace in compile time 😊
- There is no real constants in Python !!!
- Everything is variable, in fact.
- We understand the difference with their names:
 - `variableInMixUse`
 - `CONTANTS_ALL_CAPITAL`

Constants (C & Python)

- true, false
- False is 0 and true is other numbers, similar to almost all other programming languages:
 - `if (true)`
 - `if (1)`
 - `x = 5; if (x)`
 - `char answer = 'Y'; if (answer)`
 - `char answer = 'N'; if (answer)`
- True, False
- Starts and ends with double underscore (`__`) indicates Python-specific constants:
 - `__name__`
 - `__file__`
 - `__main__`



Thanks for
listening 😊

YALÇIN İŞLER

Assoc. Prof.