

How to Initialize a Variable

```
data_type (var_name);
```

```
data_type (var_name) = initial_value;
```

Functions

Syntax:

```
return_type (func_name) (datatype name1, datatype name2, ...) {
    .
    .
    return something;
}
```

```
int diff(int a, int b) {    A function that takes two integers and performs subtraction
    return a - b;
}
```

How to use Struct

```
struct (struct t_name) {    A struct is a type of pointer. Similar to making a class.
    (datatype) name1;
    (datatype) name2;
    .
};
```

```
struct struct_name (name) = { val1, val2, ..}    Initializes a struct with the following values
;
```

```
p->x;    The arrow can be used to access members of a struct. This code accesses x
        from a pointer struct called p. Equivalent to (*p).x
```

<stdio.h> and <string.h> functions

<code>strlen(string)</code>	Counts the total char in a string	<pre>char msg[] = "mes sag e"; printf("%s", strlen(msg)); // This code prints 7</pre>
<code>strcat(string1, string2)</code>	Is used to join two strings together string1 adds the text of string2 to it	<pre>char msg1[] = "My name is "; char msg2[] = "Vin"; strcat(msg1, msg2); printf("%s", msg1); //This code prints "My name is Vin"</pre>
<code>strcmp()</code>	Is used to tell if two strings are equal. Returns 0 if they are equal. Returns a non-zero value if they are not.	<pre>char msg1[] = "xyz"; char msg2[] = "xyz"; int result = strcmp (msg1, msg2); printf("%d", result); //This code prints 0</pre>
<code>strchr()</code>	Is used to find first occurrence of a char in a string	<pre>char word = "wolves"; void res = strchr (word, 's'); printf("%s", res); //This code returns a pointer to s</pre>



<stdio.h> and <string.h> functions (cont)

strstr() Is used to find first occurrence of a string in a string

```
char sent[] = "The world is cold";
void* ptr = strstr (sent, "orl");
if (sent != NULL) {
    printf("%s", ptr);
}
//This code prints if substring is found
```

strcpy() One string copies another

```
char msg[] = "Jim";
char* nmsg = malloc (strlen(msg) * sizeof(char));
strcpy(nmsg, msg);
printf("%s\n", nmsg);
//This code prints " Jim ", nmsg was changed
```

strdup Allocates and copies the string

```
char s = "some string "; char copy = strdup(s);
```

DataTypes		
char	refers to characters or strings	Ex. "car", 'c'
int	refers to an integer	Ex. 1, 2, 390
float	refers to a decimal, up to 6 digits	Ex. 1.909034
double	refers to a decimal, up to 15 digits	Ex. 1.9090341111
void	Empty	

Pointers	
Description:	A variable that stores another variables address
*	The Dereference Operator:access the value stored at the address a pointer is pointing to
&	The Address Operator: It is used to get the memory address of a variable.
<pre>int num = 6; int *ptr = &num;</pre>	Return the address of num to ptr, and then dereference it. Essentially creating a pointer

Loops	
for Loop	<pre>for (int x = 0; x < 5; x++) { /body/ };</pre>
while Loop	<pre>while (x<5) { /body/ };</pre>

How to use malloc()	
Allocates space in memory of a specific block size. Returns a void pointer if successful. Remember to free the pointer when done.	
<pre>int nums = malloc (sizeof (int) 10);</pre>	Creates an array of size 10

How to use realloc()	
Re-allocated space of a given malloc() block space, will preserve data that's already there as long the new space is not smaller	
<pre>int* arr = malloc(5 * sizeof(int)); arr = realloc(arr, 10 * sizeof (int));</pre>	The array that was created with malloc() is increased to be able to contain 5 more elements using realloc()

Arrays	
Syntax:	<i>(datatype) (var_name)[]</i>
int arr[10];	An integer array with space for 10 integers
int nums[] = {1,2};	An integer array with elements declared
int arr[5] = {0};	An integer array of all zeroes
int arr3[5] = {1, 2};	An integer array, first two elements are set, others are 0
char word[] = "Hello"	A char array which is basically a string

Format Specifiers		
%c	Used for character data	char
%d	Used for signed integer data	int
%u	Used for unsigned integer data	unsigned int
%f or %.	Used for float or double, can insert a number before "f" for precision	float or double
(num)f		
%s	Used for string data	char (string)[] or char* (string)
%p	Used for printing the address of a pointer	void *(pointer)

Importing files		
To import files use <i>#include ...</i> at the top of the file, these are .h files		
Use < ... > if importing from standard c library		
Use " ... " if importing your personal file		
#include <stdio.h>	Standard Input Output library	printf(); scanf();
#include <string.h>	A library with string manipulation functions	strlen(); strcpy(); strcat(); memcpy(); memset();
#include <stdlib.h>	Standard Library	malloc(); realloc(); free(); rand();

