

Programación Básica

Alberto Benavides

Ago - Dic 2018

4. Tipos de datos

Observaciones tarea (I)

$$930_{10} = 1110100010_2$$

Factor	Residuo		Factor	Residuo
930/2	0		29/2	1
465/2	1		14/2	0
232/2	0		7/2	1
116/2	0		3/2	1
58/2	0		1/2	1

Observaciones tarea (II)

```
#include <iostream>
using namespace std;

void main(){
    // :c
    int i;
    char c;
    [...]

    // :D
    int j;
    char d;
    [...]
}
```

Observaciones tarea (III)

```
// es std no stc
using namespace std;
[...]
```



```
cout << "aquí "  
      << "no "  
      << "hay "  
      << "saltos.";
```



```
// es endl no end1
cout << "Aquí \n"  
      << "sí." << endl;
```

Observaciones tarea (IV)

- ¡Ubicación del archivo `cpp` !
- 2482: Reenviar archivo con correcciones.

Definición de tipos de datos

- Reservan espacio en memoria
- Son diferentes
 - Información que almacenan
 - Rangos de valores válidos
- Se asignan a **variables**

```
tipoDeDato nombreDeLaVariable;
```

Repaso tipos de datos

```
int entero;  
entero = 4;  
  
char character;  
character = 'a'; // ¡Comillas simples!  
  
char cadena[10];  
// cadena = "hola"; // Da error  
cin >> cadena;  
  
// Variable vacía para funciones sin valor de retorno  
void main(){}  

```


Tipos de datos

```
long enteroLargo;
```

```
float flotante;
```

```
flotante = 1.0f; // Se debe incluir f al final
```

```
flotante = 1;
```

```
double flotanteLargo;
```

```
bool booleana;
```

```
booleana = true;
```

```
booleana = false;
```

Cadenas de caracteres

```
#include <string>
[...]  
string cadenaPlus;  
cadenaPlus = "¡No hace falta especificar longitud!";  
  
// Obtener la línea de entrada (con espacios incluidos)  
getline(cin, cadenaPlus);
```

Solucionar problema de compilación con g++ *



Conversión entre tipos de datos *

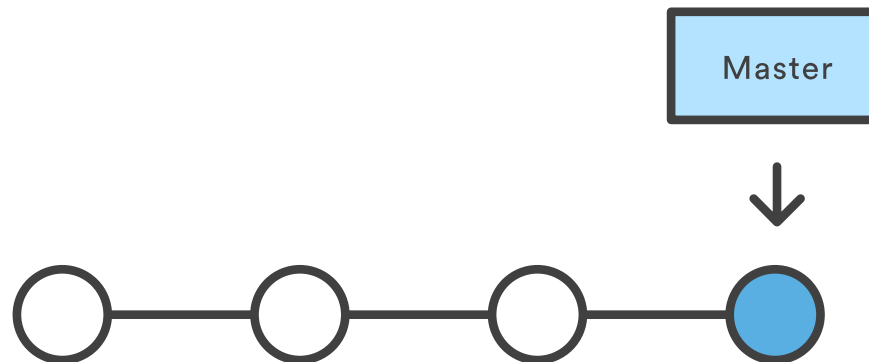
```
// char[] to int  
int i = atoi("1234");
```

```
// string to int  
int i = stoi("1234");
```

```
#include <string>  
[...]  
  
// Valor numérico a cadena  
string n = to_string(1234);  
string f = to_string(12.34f);
```

\approx Control de versiones

	
1.Encuadras objetos deseados.	1.Preparas archivos (stage)
2.Tomas foto.	2.Commit
3.Guardas en un álbum.	3.Repositorio



Instalación de Sourcetree

1. Abrir Sourcetree
 2. Tools>Options>General
 - Name: "Nombre Apellido"
 - Email: "usuario@correo.com"
-

Crear cuenta en Github

1. New repository
2. Repository name
3. Public
4. Create

Instrucciones de Sourcetree (una vez por repo)

Crear repositorio | Hacer un álbum

- File > Clone/New... > Create
 - Elegir carpeta y nombrar repositorio
 - Create
-

Añadir remoto

- Settings (engrane, parte superior derecha)
- Remotes > Add
- (Del repositorio de Github) Clone or Download > Copiar URL
- Pegar URL en URL / path
- Default remote
- OK

Instrucciones de Sourcetree (una vez por cambio)

Stage files | Preparar archivos

- (Columna izquierda) File status > Working copy
 - Seleccionar archivos > Stage [all | selected]
-

Hacer un commit | Tomar un foto

- Escribir mensaje del commit
- Commit (esq. inferior derecha)

Instrucciones Sourcetree (publicar cambios)

- Push (superior derecha)
- Elegir master
- Push

! Tarea !

- Crear un repositorio.
- Incluir una carpeta llamada "Tarea 2".
- Guardar en la carpeta "Tarea 2" el archivo `cpp` correspondiente a la tarea 2.
- Pasar la liga del repositorio de Github al correo jose.benavidesvz@uanl.edu.mx con el asunto:

PB Repositorio [Matrícula]

Fuentes

- https://www.tutorialspoint.com/cplusplus/cpp_variable_types.htm
- <http://www.cplusplus.com/doc/tutorial/variables/>