



#### Tema 4

Definición  
Ejemplos  
Captura de  
argumentos  
Ejercicios

## Tema 4: Paso de argumentos al programa

### Programación 2

Curso 2013-2014

Notas

---

---

---

---

---

---

---

---

---

---



#### Tema 4

Definición  
Ejemplos  
Captura de  
argumentos  
Ejercicios

## Índice

- 1 Definición
- 2 Ejemplos
- 3 ¿Cómo capturar los parámetros en nuestro programa?
- 4 Ejercicios

Notas

---

---

---

---

---

---

---

---

---

---



#### Tema 4

Definición

Ejemplos

Captura de  
argumentos

Ejercicios

## ¿Para qué sirven los argumentos del programa?

- Los argumentos o parámetros se usan para proporcionar información al programa (normalmente, opciones) desde la línea de comandos
- Permiten la ejecución por lotes (no interactiva) de un programa o la ejecución desde otro programa
- Es mucho más rápido (para el usuario experto) usar argumentos

Notas

---

---

---

---

---

---

---

---

---

---



#### Tema 4

Definición

Ejemplos

Captura de  
argumentos

Ejercicios

## Ejemplos de argumentos del programa

- Para ver el contenido de un directorio  

```
$ ls -l -a
```
- Para editar un programa (desde el terminal):  

```
$ joe holamundo.cc
```
- Para compilar un programa:  

```
$ g++ -Wall -g holamundo.cc -o holamundo
```

Notas

---

---

---

---

---

---

---

---

---

---

## ¿Cómo capturar los parámetros en nuestro programa?

El 'main' también es una función

```
int main(int argc, char *argv[])
{
    ...
    return 0;
}
```

- `argc` : número de argumentos (contando el nombre del programa)
- `argv` : vector de cadenas de caracteres con los argumentos

Notas

---

---

---

---

---

---

---

---

---

---

## El ejemplo clásico: 'argumentos.cc'

```
int main(int argc, char *argv[])
{
    int i;

    for (i=0;i<argc;i++)
        cout << "Arg. " << i << " : "
              << argv[i] << endl;
}
```

\$ argumentos uno dos tres

Arg. 0 : argumentos  
Arg. 1 : uno  
Arg. 2 : dos  
Arg. 3 : tres

Notas

---

---

---

---

---

---

---

---

---

---

- ... pues no lo es
- Puede haber errores (mostrar mensaje de ayuda)
- El usuario no siempre usa el mismo orden:  

```
g++ -Wall -o programa programa.cc -g
```
- Es recomendable usar una función aparte para gestionar los argumentos

---

---

---

---

---

---

---

---

---

---

### Ejercicio 1

Implementa un programa con la siguiente función:

```
int NumeroPrimo(int n);
```

que devuelve el *n*ésimo número primo. El programa debe imprimir números primos por pantalla con las siguientes opciones:

- L      imprimir cada número en una línea distinta  
         (por defecto se imprimen todos en una línea)
- N *n*   imprimir los *n* primeros números primos (por defecto 10)

Ejemplos de ejecución:

```
$ primos -N 5  
1 2 3 5 7
```

```
$ primos -N -L 5  
Error: primos [-N n] [-L]
```

---

---

---

---

---

---

---

---

---

---