Tipado débil y tipado fuerte: diferencias y explicación - Parzibyte's blog

parzibyte

Introducción

Tal vez hemos escuchado que algunos lenguajes de programación son de tipado fuerte y otros de tipado débil. Hay una gran diferencia entre ambos tipos. Normalmente el tipado débil es el más criticado de todos, pero también es querido por muchos. Aquí daré una pequeña explicación.

Para comenzar, debemos saber que el tipado se refiere a cómo declaramos los tipos de variables. Por ejemplo, algunas las declaramos como enteras, algunas otras como cadena, flotantes, etcétera. Y en algunos lenguajes, no necesitamos declarar el tipo, pues éste se adivina.

Por otro lado, el tipado fuerte no permite hacer operaciones entre objetos de distintos tipos. No podemos sumar una cadena mas un entero. En cambio, en los débilmente tipados sí.

La mayoría de veces, el tipado débil es en donde no indicamos el tipo de variable al declararla. La verdadera diferencia es que podemos asignar, por ejemplo, un valor entero a una variable que anteriormente tenía una cadena.

También podemos operar aritméticamente con variables de distintos tipos. Por ejemplo, sumar “x” + 5.

Ventajas

- Nos olvidamos de declarar el tipo
- Podemos cambiar el tipo de la variable sobre la marcha. Por ejemplo, asignarle un **string** a un **int**
- Escribimos menos código

Desventajas

- Al hacer operaciones, a veces éstas salen mal. Por ejemplo, puede que intentemos sumar 500 + “400.00” + 10, cosa que será errónea
- Especialmente en Javascript, he notado que hay **errores al comparar números que creemos que son números**, pero no lo son. Por ejemplo, algo como “20” > “100” **dará como resultado true**, ya que son comparados como cadena, no como números.
- Hay que castear muchas veces. En ocasiones, tendremos que castear forzosamente las variables para que se comporten como queremos y no generen errores como los mencionados arriba.
- Código menos expresivo. Al declarar los argumentos de una función no sabemos si ésta espera un flotante, un entero, un string, etcétera. Tenemos que ir a la función, ver lo que hace e inferir el tipo de variable que espera
- Inseguridad: existe la posibilidad de que un atacante descubra una vulnerabilidad en donde nosotros esperemos una variable de determinado tipo pero se reciba otra

Lenguajes que lo usan

- PHP
- Javascript

Ejemplo

Veamos lo que pasa en JavaScript cuando hacemos la siguiente operación:

```
let resultado = "x" + 5;
```

En un lenguaje fuertemente tipado daría un error, pero en JavaScript no pasa nada:

```
> let resultado = "x" + 5;
< undefined
> resultado
< "x5"
```

Tipado fuerte

Aquí es en donde indicamos el tipo de dato al declarar la variable. Dicho tipo no puede ser cambiado nunca. Y no podemos operar entre distintos tipos.

Ventajas

- Código expresivo: ahora sí sabremos de qué tipo espera un argumento una función
- Menos errores: Nos olvidaremos de ver el tipo de variable antes de hacer operaciones con ésta

Desventajas

- Escribir más codigo: tenemos que declarar el tipo de variable al declararla

Lenguajes que lo usan

Por mencionar algunos...

- C
- C#
- Java
- Ruby
- Python

Ejemplo

Intentemos realizar la operación de “x” + 5 en Python, y veamos lo que pasa:

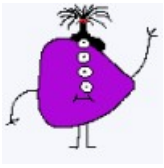
```
>>> resultado = "x" + 5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not int
>>>
```

Conclusión

Es un poco difícil determinar los de tipado débil y los de tipado fuerte. Los de tipado fuerte son más seguros, pues no permiten hacer operaciones con variables de distintos tipos.

Personalmente preferiría que todos fueran de tipado fuerte. Si no nos queda otra opción, simplemente hay que castear o comprobar el tipo cada vez que trabajemos con una variable. Afortunadamente tenemos funciones como gettype, settype, typeof, etcétera.

Además, algunos lenguajes (como PHP) están introduciendo un tipado fuerte al declarar funciones. De todos modos, al final todo depende del programador, pues éste se encarga de que el código funcione.



parzibyte

Programador freelancer listo para trabajar contigo. Aplicaciones web, móviles y de escritorio. PHP, Java, Go, Python, JavaScript, Kotlin y más :) https://parzibyte.me/blog/software-creado-por-parzibyte/