

## CURSO DE FUNDAMENTOS DE JAVA

# PALABRA `this` EN JAVA



Por el experto: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de la palabra `this` en Java.

¿Estás listo? Ok, ¡Vamos!

## PALABRA this EN JAVA

### PALABRA this

- Es una referencia implícita al objeto que se está ejecutando.
- Es útil para evitar la ambigüedad entre las variables de clase y las locales
- Permite a un objeto enviarse él mismo como parámetro

### CURSO DE FUNDAMENTOS DE JAVA

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hemos visto que la palabra `this` nos permite acceder al objeto que se está ejecutando actualmente, sin embargo vamos a dedicar algunos minutos más a este tema para comentar algunos detalles más.

La palabra `this` es un operador y básicamente nos permite acceder al objeto que se está ejecutando en cualquier método Java, sin embargo esto en un principio puede ser algo confuso.

Tenemos dos tiempos en Java según hemos visto, el tiempo de compilación, que es cuando estamos creando nuestro código, y el tiempo de ejecución que es cuando nuestro programa está en funcionamiento.

Al momento de crear nuestro código debemos tener presente que no se está ejecutando ningún código, pero que en un futuro es precisamente lo que sucederá, sobre todo cuando estamos creando nuestras clases. Por ello, el operador `this` se tiene que pensar en términos de ejecución de nuestro código, es decir, cuando se crea un objeto de la clase que estamos codificando.

El operador `this` es único para cada objeto, y siempre corresponderá al código del objeto que se esté ejecutando.

Otra cosa importante de mencionar del operador `this` es que esta palabra únicamente se puede utilizar dentro de un objeto creado a partir de la clase Java, más adelante veremos lo que significa el contexto estático, pero podemos adelantar que la palabra `this` no se puede utilizar en un contexto estático, ya que en el contexto estático aún no se han creado los objetos, y por lo tanto la palabra `this` no tiene utilidad si no se ha creado el objeto al cual debe apuntar.

# USO PALABRA this EN JAVA

## Creación de la clase

Tiempo  
Compilación

```
class Persona{
    String nombre;

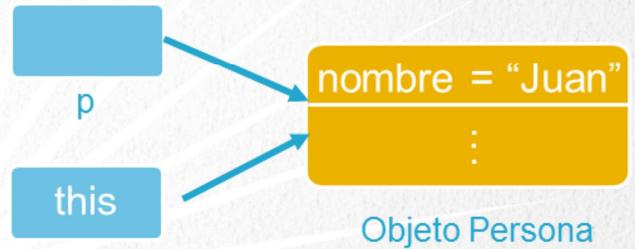
    Persona(String nombre){
        this.nombre = nombre;
    }
}
```

Efecto en la Memoria

Tiempo  
Ejecución

## Extracto Clase de Prueba

```
public static void main(String[] args) {
    Persona p = new Persona("Juan");
}
```



## CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Uso de los usos más comunes de la palabra `this`, como ya hemos comentado, es para evitar la ambigüedad entre los atributos de las clases (miembros de clase) y las variables locales o argumentos de un método.

En el ejemplo podemos observar cómo estamos declarando una clase llamada `Persona`, la cual tiene un constructor que recibe un atributo llamado `nombre`, el cual es idéntico al nombre del atributo de la clase. Para hacer la diferencia entre ambos nombres, podemos utilizar la palabra `this` la cual hace referencia al atributo de la clase, de esta manera podemos nombrar de manera idéntica los argumentos de un método y diferenciar los atributos de la clase por medio del operador `this`.

En la figura podemos observar como tanto la variable `p` como el operador `this` apuntan al mismo objeto al momento de llamarse el constructor de la clase. Una vez que termina de ejecutarse el constructor de la clase el operador `this` apunta a otro objeto, y si de nueva cuenta se vuelve a ejecutar algún método de la clase `Persona`, el operador `this` apuntará al objeto creado, es decir que el operador `this` siempre apuntará al objeto que se está ejecutando actualmente.

## USO PALABRA this COMO REFERENCIA

### Uso palabra this como referencia:

```

1 public class PalabraThis {
2     public static void main(String[] args) {
3         Persona p = new Persona("Juan");
4     }
5 }
6
7 class Persona {
8     String nombre; //atributo de la clase
9
10    Persona(String nombre) {
11        this.nombre = nombre; //this es el objeto Persona (actual)
12
13        //Imprimimos el objeto persona
14        Imprimir i = new Imprimir();
15        i.imprimir(this); //this es el objeto Persona (actual)
16    }
17 }
18
19 class Imprimir {
20
21    public void imprimir(Object o) {
22        System.out.println("Imprimir parametro: " + o); //el parametro es el objeto persona
23        System.out.println("Imprimir objeto actual (this): " + this); //this es el objeto imprimir (actual)
24    }
25 }

```

En este ejemplo podemos observar a más detalle el uso de la palabra this. Estamos creando varias clases y varios objetos para que podamos observar cómo la palabra this va cambiando según el objeto actual que se está ejecutando.

En la línea 3 creamos un objeto de tipo Persona. Hasta este punto el apuntador this no ha sido creado ni apunta a ningún objeto. Es sólo hasta que se ejecuta la línea 3 que se crea el objeto de tipo persona y entonces se crea el apuntador this y apunta al objeto Persona recién creado.

Sabemos que el paso inmediato en la creación de un objeto en Java es llamar al constructor de la clase, por ello como siguiente paso se ejecuta la línea 10, en la cual recibimos en la variable local de tipo nombre el valor de Juan, pero para poder diferenciar entre el argumento y el atributo de la clase utilizamos por primera vez la palabra this, de esta manera hacemos referencia al atributo de la clase.

Posteriormente creamos un nuevo objeto de la clase Imprimir, pero no es hasta que se ejecuta el método imprimir que el apuntador this cambiará de apuntar al objeto Persona al objeto Imprimir. Por ello en la línea 15 pasamos como parámetro el objeto Persona actual utilizando la palabra this, y cuando se ejecuta la línea 21 ocurren dos cosas importantes, por un lado el operador this deja de apuntar al objeto persona y el argumento que recibe el método imprimir es objeto Persona puesto que this en la línea 15 aún apuntaba al objeto persona.

Todas las clases en Java heredan de la clase Object, y aunque lo veremos en otro tema, la clase Object nos sirve como una clase comodín para poder recibir cualquier tipo Java como argumento, en este caso la variable Object creada almacena la referencia de la clase Persona creada anteriormente, y cuya referencia fue enviada en la línea 15 por medio del apuntador this.

Por ello en la línea 22 se imprime un objeto de tipo Persona, y en la línea 23 se imprime un objeto de tipo Imprimir, ya que recordemos que this ahora está apuntando al objeto que se está ejecutando actualmente, el cual es de tipo Imprimir.

Hay más cosas que podríamos hacer para mejorar este código, sin embargo necesitamos de más lecciones para poder mejorar este código, por esta razón vamos a dejar lo más simple posible este código, con el objetivo de entender cómo va cambiando el apuntador this según la ejecución de nuestro código.

## EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Uso de la palabra this en Java.

**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# CURSO ONLINE

# FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



**CURSO DE FUNDAMENTOS DE JAVA**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- ✔ Lógica de Programación
- ✔ Fundamentos de Java
- ✔ Programación con Java
- ✔ Java con JDBC
- ✔ HTML, CSS y JavaScript
- ✔ Servlets y JSP's
- ✔ Struts Framework
- ✔ Hibernate Framework
- ✔ Spring Framework
- ✔ JavaServer Faces
- ✔ Java EE (EJB, JPA y Web Services)
- ✔ JBoss Administration
- ✔ Android con Java
- ✔ HTML5 y CSS3

#### Datos de Contacto:

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)

