



THIS



Cuando se llama a un método, se pasa automáticamente un argumento implícito que es una referencia al objeto invocado (es decir, el objeto sobre el que se llama el método). Esta referencia se llama `this`. Para comprender esto, considere primero un programa que crea una clase llamada `Potencia` que calcula el resultado de un número elevado a una potencia entera:

```
//Un ejemplo del uso de this
```

```
class Potencia {  
    double b;  
    int e;  
    double valor;
```



```
Potencia(double base, int exp){  
    b=base;  
    e=exp;  
    valor=1;  
  
    if (exp==0) return;  
    for ( ; exp>0; exp--) valor = valor * base;  
}  
  
double get_potencia(){  
    return valor;  
}  
}  
  
class DemoPotencia {  
    public static void main(String[] args) {  
  
        Potencia x=new Potencia(4.0,2);  
        Potencia y=new Potencia(2.5,1);  
        Potencia z=new Potencia(2.7,2);  
  
        System.out.println(x.b+ " elevado a la potencia de  
        "+ x.e+", es igual a: "+x.get_potencia());  
    }  
}
```



```
System.out.println(y.b+ " elevado a la potencia de  
"+ y.e+", es igual a: "+y.get_potencia());
```

```
System.out.println(z.b+ " elevado a la potencia de  
"+ z.e+", es igual a: "+z.get_potencia());
```

```
}
```

```
}
```

Salida:

4.0 elevado a la potencia de 2, es igual a: 16.0

2.5 elevado a la potencia de 1, es igual a: 2.5

2.7 elevado a la potencia de 2, es igual a:
7.2900000000000001

Uso de this en Java

Como sabe, dentro de un método, se puede acceder directamente a los otros miembros de una clase, sin ningún objeto o calificación de clase. Por lo tanto, dentro de `get_potencia()`, la declaración `return valor;` significa que se devolverá la copia del valor asociado con el objeto invocado. Sin embargo, la misma declaración también se puede escribir así:

```
return this.valor;
```

Aquí, `this` se refiere al objeto sobre el que se llamó `get_potencia()`. Por lo tanto, `this.valor` se refiere a la copia del valor de ese objeto. Por ejemplo, si `get_potencia()` se hubiera invocado en `x`, `this` en la declaración anterior se habría estado refiriendo a `x`. Escribir la declaración sin usar `this` es solo una abreviatura.

A continuación, se muestra toda la clase `Potencia` escrita usando la referencia `this`:

```
class Potencia {  
  
    double b;
```

```
int e;

double valor;

Potencia(double base, int exp){

    this.b=base;

    this.e=exp;

    this.valor=1;

    if (exp==0) return;

    for ( ; exp>0; exp--) this.valor = this.valor *
base;

}

double get_potencia(){

    return this.valor;

}

}
```

En realidad, ningún programador de Java escribiría una clase Potencia como se acaba de mostrar porque no se gana nada, y la forma estándar es más fácil. Sin embargo, esto tiene algunos usos importantes. Por ejemplo, la sintaxis de Java permite que el nombre de un parámetro o una variable local sea el mismo que el nombre de una variable de instancia.

Cuando esto sucede, el nombre local oculta la variable de instancia. Puede obtener acceso a la variable de instancia oculta haciendo referencia a ella a través de `this`. Por ejemplo, la siguiente es una forma sintácticamente válida para escribir el constructor de Potencia().

```
Potencia(double b, int e){  
    this.b=b;  
    this.e=e;  
    valor=1;  
  
    if (e=0) return;  
    for ( ; e>0; e--) valor = valor * b;  
}
```

En esta versión, los nombres de los parámetros son los mismos que los nombres de las variables de instancia, por lo tanto, los ocultan. Sin embargo, this se usa para “descubrir” las variables de instancia.