



ORACLE

Academy



Java Foundations

3-1

¿En Qué Consiste una Variable?

ORACLE
Academy



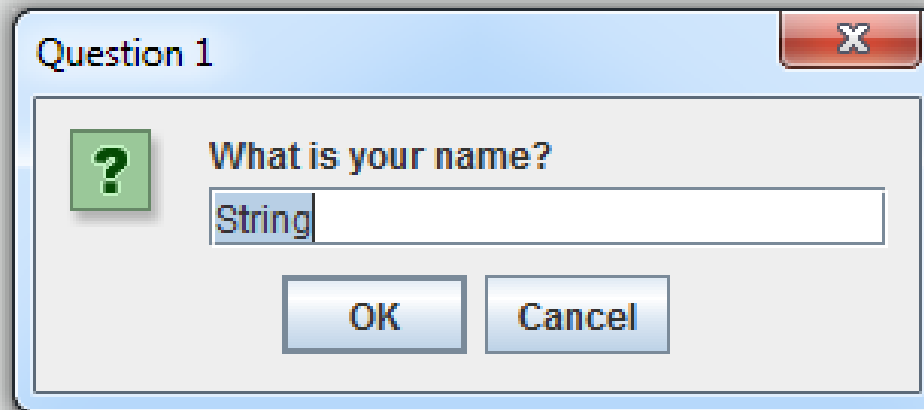
Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Comprender las ventajas de las variables
 - Identificar cuatro tipos principales de variables:
 - (boolean, int, double, String)
 - Declarar y asignar valores a las variables
 - Nombrar variables según las convenciones



Ejercicio 1

- Ejecute JavaLibs.jar
- Piense en los tipos de datos que pide este programa



El conjunto de problemas 3 tiene por objetivo volver a crear este programa con su propia historia En esta sección se explica todo lo necesario para que pueda crear este programa

¿En qué consiste una variable?

- Fijémonos en la variable x de la ecuación
- Podemos asignar cualquier valor a x

$$y = -2x + 5$$

$$x = 0$$

$$y = -2 \times 0 + 5$$

$$y = 0 + 5$$

$$y = 5$$

$$x = 2$$

$$y = -2 \times 2 + 5$$

$$y = -4 + 5$$

$$y = 1$$

¿Qué Es una Variable en Java?

- Del mismo modo, podemos asignarle valores a las variables de Java
 - Al escribir la línea `String x = "Alex"`, es como si le estuviéramos asignando a la variable `x` el valor `"Alex"`
 - Escribir `"My name is " + x` equivale a escribir `"My name is Alex"`

```
String x = "Alex";  
System.out.println("My name is " + x);
```



`"My name is Alex"`

Desventajas de No Utilizar Variables

- El código no es flexible
- Para cambiar el nombre “Alex,” habría que hacer muchos cambios por todas partes:
 - La edición se convierte en una tarea tediosa
 - Se corre el riesgo de que acabe faltando un “Alex”

```
System.out.println("My name is Alex");
System.out.println("Alex is so cool!");
System.out.println("Hooray Alex!");
System.out.println("Please enjoy Alex Appreciation "
    + "Day! My name is Alex. I know how excited "
    + "everyone is to start appreciating Alex on Alex"
    + "Appreciation Day! Alex, Alex, Alex! Yay "
    + "Alex!!! That's me! Alex is the best date ever!");
```




Ventajas de utilizar Variables

- El código es flexible
 - Se pueden recordar y manipular los valores
- Para sustituir el nombre “Alex,” solo tiene que hacer un cambio:
 - La edición se convierte en una tarea fácil
 - No se corre el riesgo de que acabe faltando un “Alex”

```
String x = "Sam";
System.out.println("My name is " + x);
System.out.println(x + " is so cool!");
System.out.println("Hooray " + x + "!");
System.out.println("Please enjoy " + x + " Appreciation "
+ "Day! My name is " + x + ". I know how excited "
+ "everyone is to start appreciating " + x
+ " on " + x + "Appreciation Day! " + x + "," + x + ","
+ x + "! Yay " + x + "!!! That's me! " + x
+ " is the best date ever!");
```


Más Ventajas de Utilizar Variables

- Permite manipular valores muchas veces de varias formas:
 - Cambiando directamente los valores usted mismo (como se muestra a continuación)
 - Cambiando los valores calculados mediante programación
 - Realizando cambios en función de las entradas de los usuarios

```
5    String x = "Alex";
6    x = "Sam";
7    x = "Nicky";
8    x = "Mystery Date";
9
10   "backwards" = x;    //Can't do this
```

Ejercicio 2

- Cree un nuevo proyecto y agréguele el archivo `Variables02.java`
- Realice los pasos del ejercicio
- Ejecute el programa después de cada paso y mire qué sucede
- El programa debería generar los siguientes resultados:

– Después del paso 1)

<code>puppy</code> <code>puppy</code>
--

– Después del paso 2)

<code>kitty</code> <code>kitty</code>
--

– Después del paso 3)

<code>kitty</code> <code>bunny</code>
--

Naturaleza de los Programas Línea a Línea

- A partir de la línea 8, x siempre equivale a "kitty", hasta la...
- Línea 14 en adelante donde x siempre equivale a "bunny"

```
7 public static void main(String[] args) {  
8     String x = "kitty";  
9     System.out.println(x);           //prints "kitty"  
10    //System.out.println(x);  
11    //System.out.println(x);  
12    System.out.println(x);           //prints "kitty"  
13  
14    x = "bunny";  
15    //System.out.println(x);  
16    //System.out.println(x);  
17    System.out.println(x);           //prints "bunny"  
18    //System.out.println(x);  
19  
20 }  
21 }
```




Muchos Tipos de Variables

- Hay variables para muchos tipos diferentes de datos en Java
- Estas son las variables que ya hemos visto:

Tipo	Palabra Clave	Valores de ejemplo
Boolean	<code>boolean</code>	<code>true</code> , <code>false</code>
Integer	<code>int</code>	1, -10, 20000, 123_456_789
Double	<code>double</code>	1,0, -10,0005, 3,141
String	<code>String</code>	"Alex", "I ate too much dinner."

Declaración de Variables

- Java es un "lenguaje de tipo específico"
 - Debe declarar el tipo de dato que la variable manejará mediante palabras clave




```
boolean bool;  
int x;  
double y;  
String z;
```

- Una vez que declare una variable...
 - Dicha variable comenzará a existir
 - No es necesario volver a declararla

Opciones para Declarar y Asignar Valores

- Declarar y asignar variable en una sola línea

tipo *nombre* *valor*

`boolean bool = true;`

- Declarar una variable en una línea y asignar un valor más adelante

```
boolean bool;
```

```
bool = true;
```

Asignación de Valores Incorrectos

- Los valores asignados deben ser apropiados para el tipo de dato que haya declarado



```
int x = 3;
```



```
int z = "Puppies!";
```


Valores Matemáticos Inapropiados

- Podemos asignar cualquier valor numérico a x
- No podemos asignar un valor de cadena a x
 - ¡Esto no tendría sentido!

$$y = -2x + 5$$

$x = \text{"Puppies!"}$

$y = -2(\text{"Puppies!"}) + 5$

$y = ???$



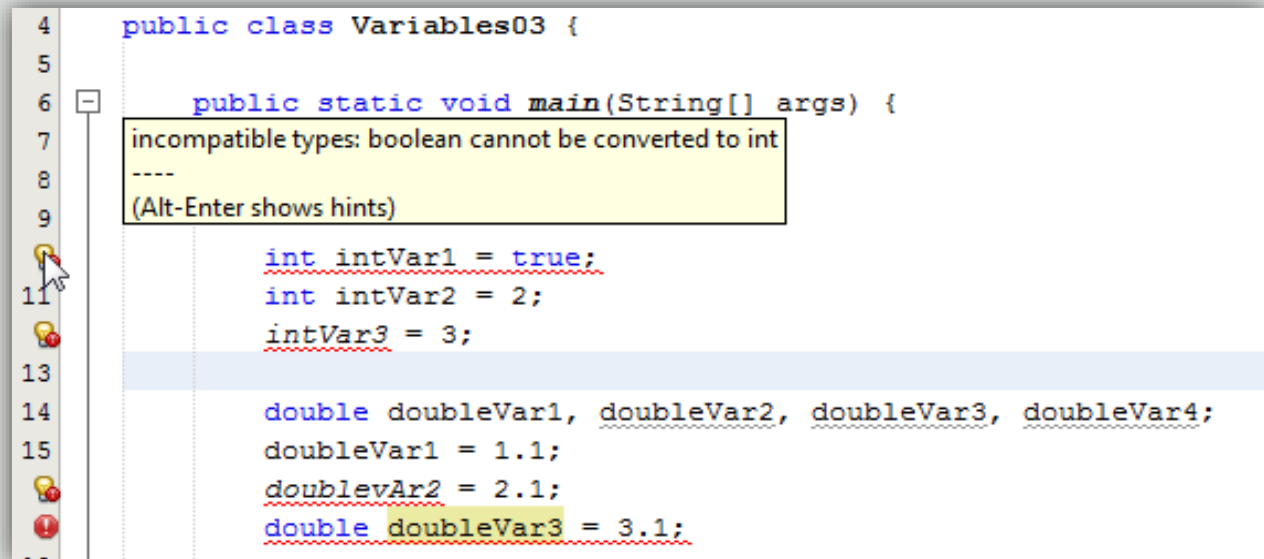
Ejercicio 3, Parte 1

- Cree un nuevo proyecto y agréguele el archivo `Variables03.java`
 - Este programa presenta seis errores
 - ¿Puede solucionar estos errores para que el programa genere el siguiente resultado?

```
bool = true
intVar1 = 1
intVar2 = 2
intVar3 = 3
doubleVar1 = 1.1
doubleVar2 = 2.1
doubleVar3 = 3.1
doubleVar4 = 4.1
stringVar1 = 11
stringVar2 = 22
```

Ejercicio 3, Indicaciones 1

- Su IDE subraya el código problemático
 - Mantenga el cursor sobre el código o icono que aparecen en el margen izquierdo para obtener más información
 - Su IDE puede hacer sugerencias sobre posibles soluciones
 - Haga clic en el icono que aparece en el margen izquierdo



```
4 public class Variables03 {  
5  
6     public static void main(String[] args) {  
7         incompatible types: boolean cannot be converted to int  
8         ----  
9         (Alt-Enter shows hints)  
10  
11         int intVar1 = true;  
12         int intVar2 = 2;  
13         intVar3 = 3;  
14  
15         double doubleVar1, doubleVar2, doubleVar3, doubleVar4;  
16         doubleVar1 = 1.1;  
17         doubleVar2 = 2.1;  
18         double doubleVar3 = 3.1;  
19     }  
20 }
```

Ejercicio 3, Indicaciones 2

- Las soluciones sugeridas por su IDE a veces no son adecuadas
 - No confíe por completo en las soluciones que sugiera su IDE
- No subestime su propia capacidad para solucionar problemas



Errores con Variables

- Asignar valores inapropiados a un tipo de variable

```
int intVar1 = true;
```

- Olvidarnos de declarar un tipo de variable

```
intVar3 = 3;
```

- Escribir mal una variable

```
double doubleVar2;  
doublevAr2 = 2.1;    //Java is case-sensitive
```

Errores con Variables

- Declarar dos veces la misma variable

```
double doubleVar3;  
double doubleVar3 = 3.1;
```

- Olvidarnos de asignar un valor antes de utilizar una variable

```
double doubleVar4;  
System.out.println(doubleVar4);
```

La asignación de un valor inicial a la variable se denomina "inicialización"

Puede que Haya Observado que...

- Es posible declarar muchas variables en una sola línea

```
double doubleVar1, doubleVar2, doubleVar3;
```

- Es posible asignar valores al declarar muchas variables

```
double doubleVar1, doubleVar2, doubleVar3 = 3.1;
```

- Es una cuestión de gusto personal...

- Declarar cada variable en líneas separadas
- Declarar todas las variables de un tipo determinado en una sola línea



Nomenclatura Inadecuada para Variables

- Puede ponerle prácticamente cualquier nombre a una variable

```
int dsfdsfspoop = 20;    //Ha ha!
```

- Esto puede parecer muy divertido, pero...
- ¿Comprenderá usted o un amigo qué representa dsfdsfspoop cuando lea el código?
- Se suele desaconsejar el uso de nombres demasiado pequeños

```
int x = 20;
```

- Resultan útiles para realizar pruebas...
- Y se encuentran comúnmente en pequeños bucles (que trataremos más adelante), pero...
- ¿Comprenderá usted o un amigo qué representan los datos x cuando lea el código?



Nomenclatura Muy Inadecuada para Variables

- Dos variables no pueden tener el mismo nombre

```
int x = 20;  
double x = 22.0;  
System.out.println(x); //Which x?
```

- Las variables no puede empezar por números

```
boolean 1337Hacker = true;
```

- No se pueden utilizar palabras clave como nombres de variables

```
int continue = 20;
```

- Las palabras clave se vuelven azules en NetBeans
- Las palabras clave tienen un significado especial en Java

Reglas de Nomenclatura de Variables



- Empezar cada variable por una letra minúscula Las siguientes palabras deben tener la inicial mayúscula:
 - myVariable
- Seleccionar nombres que sean nemotécnicos y que indiquen al observador casual la intención de la variable
- Recuerde que...
 - Los nombres son sensibles a mayúsculas y minúsculas
 - Los nombres no pueden incluir espacios en blanco

```
int studentAge = 20;  
String myCatchPhrase = "Enjoy Alex Appreciation Day!";
```

Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Comprender las ventajas de las variables
 - Identificar cuatro tipos principales de variables:
 - (boolean, int, double, String)
 - Declarar y asignar valores a las variables
 - Nombrar variables según las convenciones





ORACLE

Academy

