



Java: Clase String

Facultad de Ciencias

Diana Isabel Ramírez García

Introducción a Ciencias de la Computación 2021-1

Recordemos

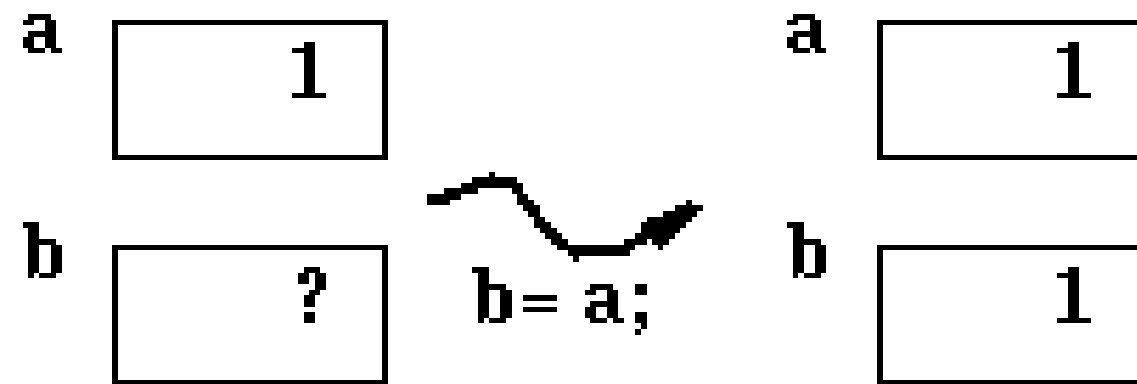
Contamos con los tipos primitivos para manipular los datos más simples, sin embargo Java cuenta **referencias** para manipular objetos.

Una referencia almacena la dirección en donde está guardado el objeto, más **no el objeto**.

Diferencia en Asignación (en memoria)

```
int a=1;  
int b;
```

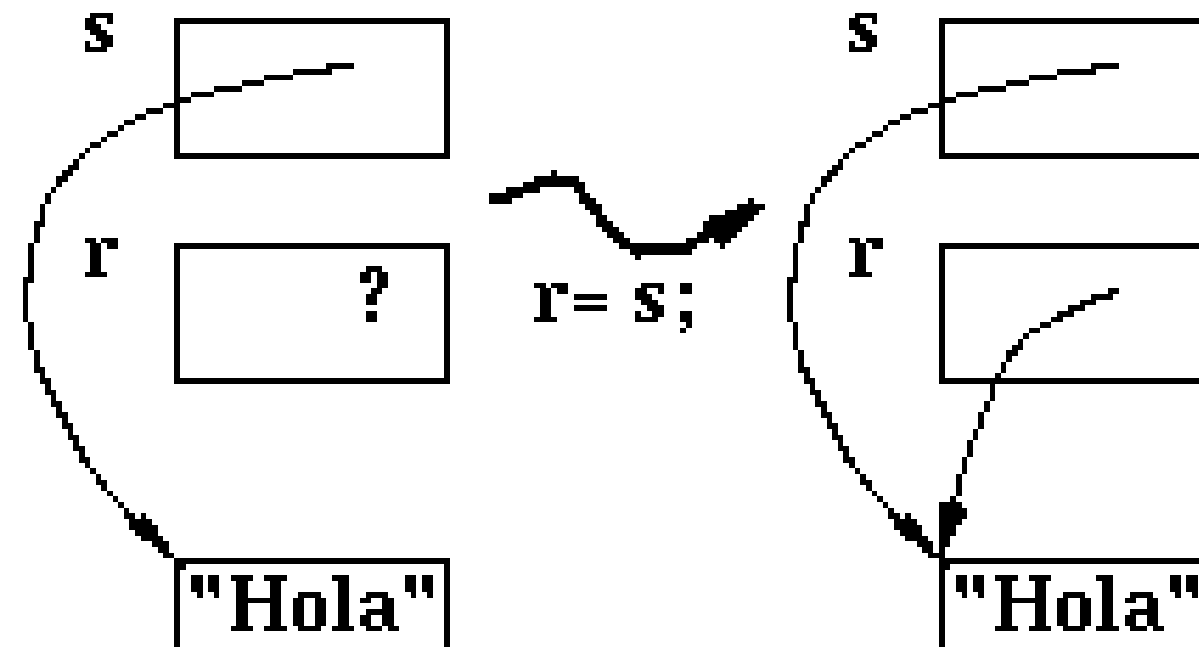
```
b = a;
```



Asignacion de variables de tipo primitivo

```
String s = "Hola";  
String r;
```

```
r = s;
```



Referencias

- Este tipo de dato únicamente cuenta con una literal llamada **null**.
- **null** nos indica que la variable no apunta a ninguna dirección.

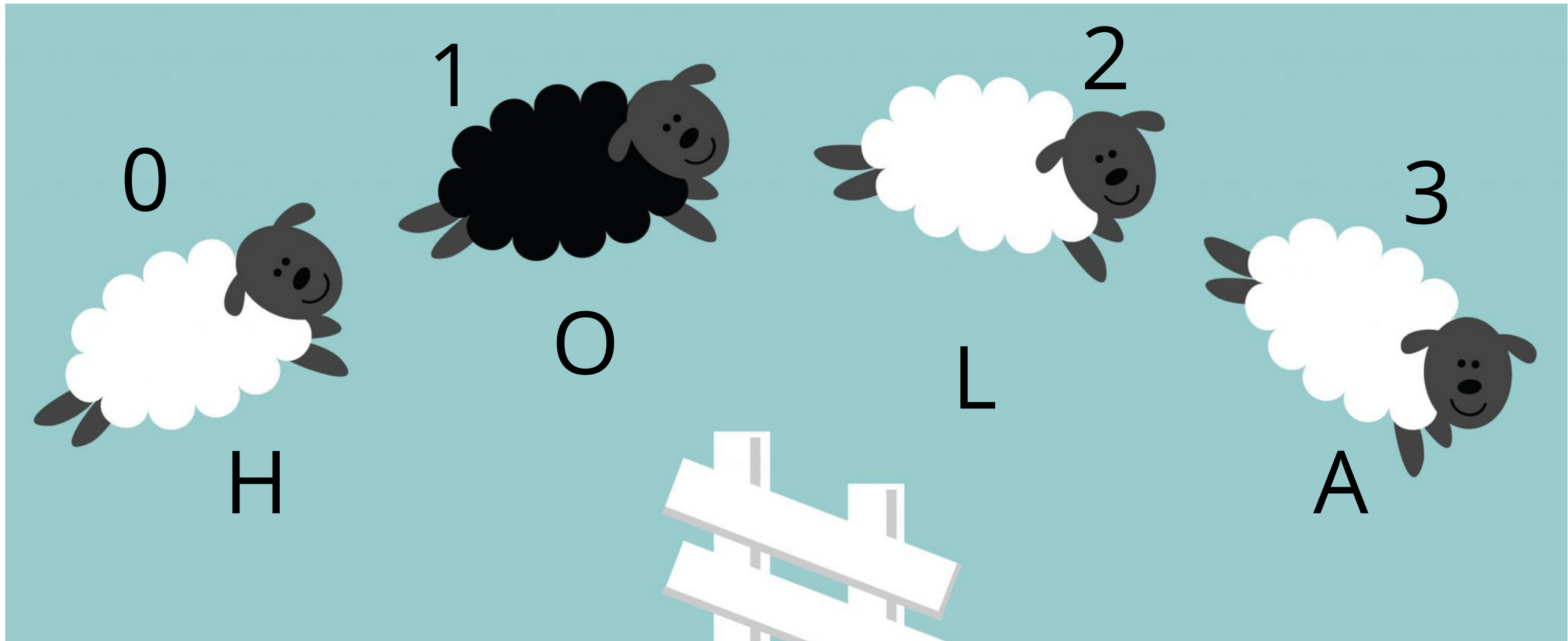
Clase String

- Las cadenas de caracteres esta definidas dentro de la clase String, implementada en Java.
- Dentro de la clase se tienen **métodos** los cuales definen el **comportamiento** de esta.

Métodos Clase String

Sintaxis	Descripción del resultado
<code>charAt(int)</code>	Devuelve el carácter de la posición indicada.
<code>equals(String)</code>	Devuelve true si la cadena pasada como parámetro es igual a la cadena que llama al método. Devuelve false en caso contrario.
<code>indexOf(String)</code>	Devuelve la posición en donde empieza la subcadena pasada como parámetro.
<code>length()</code>	Devuelve la cantidad de caracteres que tiene la cadena.
<code>substring(int, int)</code>	Devuelve la cadena formada por los caracteres que están desde la posición indicada en el primer parámetro hasta uno antes de la indicada en el segundo parámetro.
<code>toLowerCase()</code>	Devuelve la cadena en minúsculas.
<code>toUpperCase()</code>	Devuelve la cadena en mayúsculas.
<code>trim()</code>	Devuelve la cadena sin espacios en blanco al final.

Contar Posiciones



Longitud> **4**

Creación de Strings

- Podemos crear Strings de 2 maneras:
 - String literal = "hola";
 - String objeto = new String("hola");
- Ambos son objetos sin embargo los almacena diferente en memoria.
- La palabra reservada **new** la utilizamos para crear objetos.

equals() vs ==

- equals() es un método y "==" un operador
- equals() compara el contenido mientras que "==" compara si apuntan al mismo directorio.
- **Utilizar el método equals() de la clase para evitar incongruencias.**

Bibliografía

- López, A.. (2014). Introducción al desarrollo de programas con Java. México: Las prensas de Ciencias.