Strings

String is a datatype which can store collection of characters.

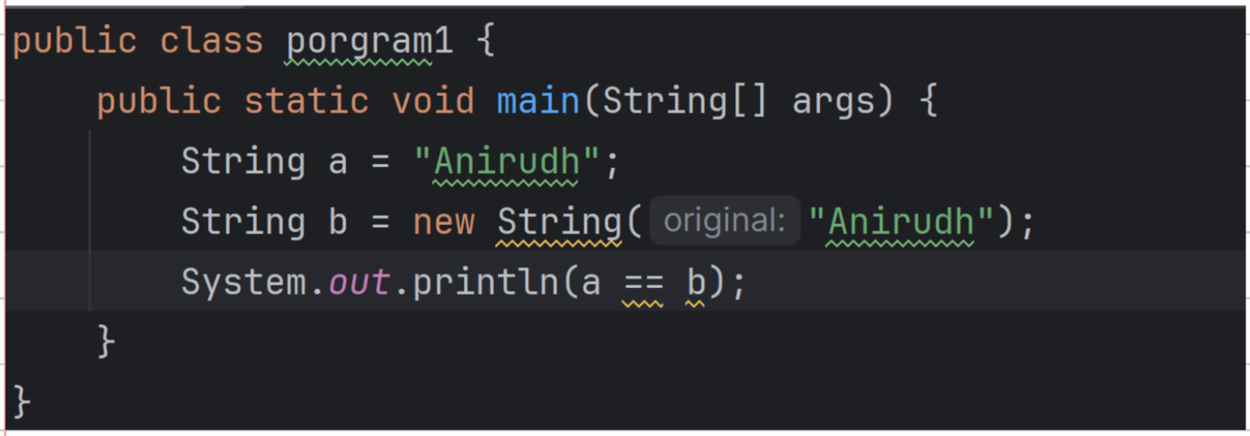
* String is a class in Java that can store collection of characters.
Everything that starts with capital letter are considered as class.

* When we declare a variable of String data type we are actually creating a object of String class.

* Strings are immutable.

Comparision of Strings:

* In Strings "==" operator will check if both the variable are pointing to same object or not



```
public class porgram1 {
    public static void main(String[] args) {
        String a = "Anirudh";
        String b = new String( original: "Anirudh");
        System.out.println(a == b);
    }
}
```

O/P: false

Despite of containing same string we will get false if we execute the above code.
We will get false because "==" will compare two strings and returns true only if the below condition satisfy.
- If both the variables are having same value / string.
- If both the variables are pointing to the same object.

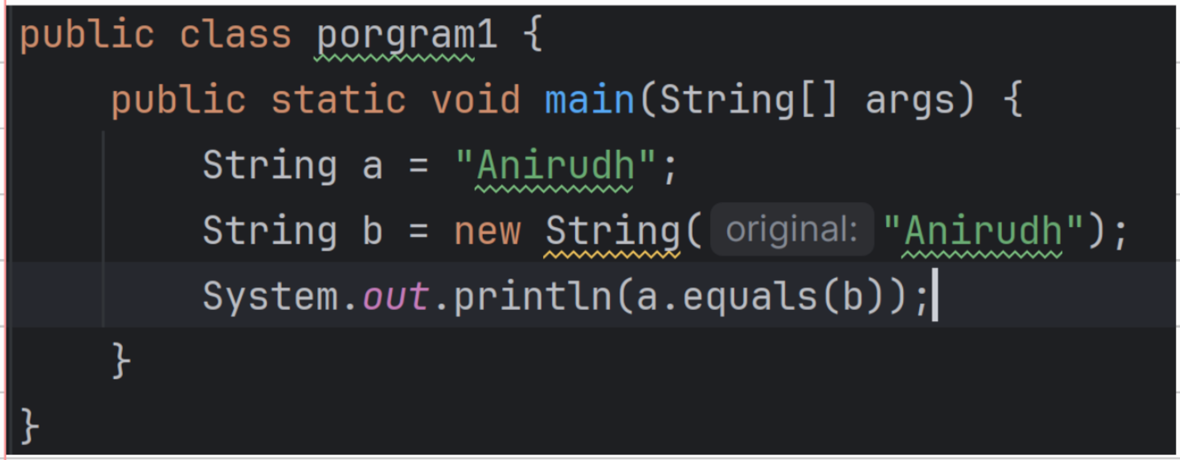
Comparision of only String values but not to which object they belong to:

If you need to compare only the values of the two strings then you can use ".equals()" method.

Syntax:

Variable_1 . equals (variable_2);

.equals() will only check if both the strings have same value it wont check if they belong to same object or not.



```
public class porgram1 {
    public static void main(String[] args) {
        String a = "Anirudh";
        String b = new String( original: "Anirudh");
        System.out.println(a.equals(b));
    }
}
```

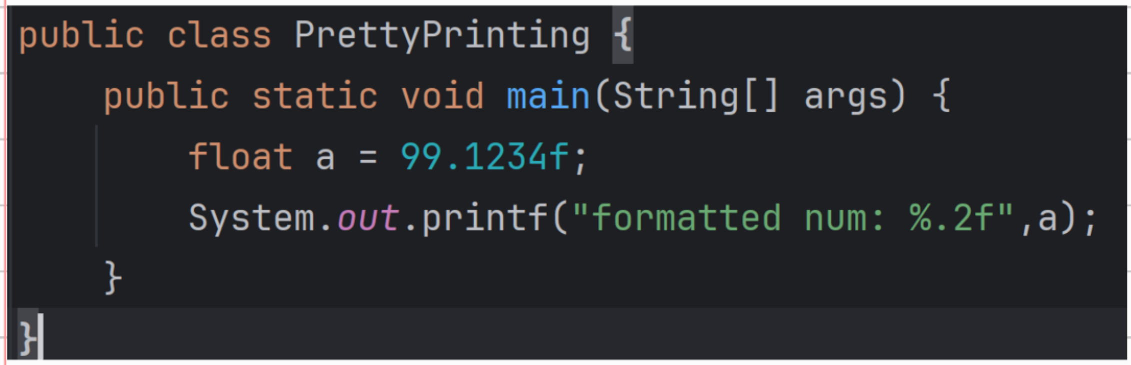
Accessing string elements:

To access individual char we use .get() method within the paranthesis we pass the index.

a.get (0);

Note: In java Strings are always represented using " " (Double quotes).
In java Char are always represented using ' ' (single quotes).
You cant represent a string in Single quotes.
Similarly you cant represent a char in double quotes.

Pretty Printing:



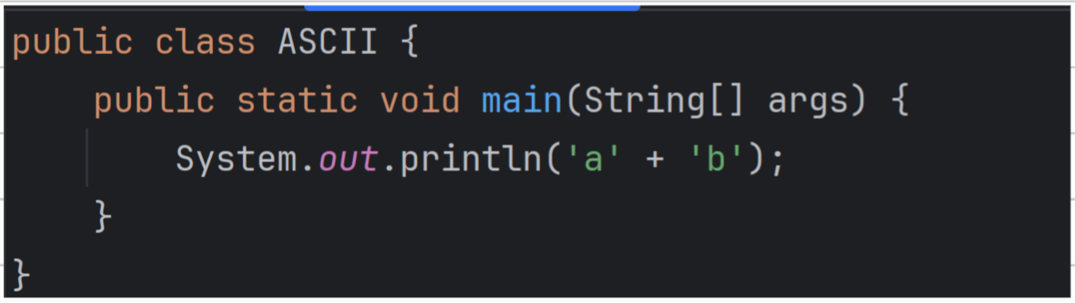
```java
public class PrettyPrinting {
    public static void main(String[] args) {
        float a = 99.1234f;
        System.out.printf("formatted num: %.2f",a);
    }
}
```

Notice that in above code we have used printf not println.

To format a number we can use % as place holder & followed by . and no. of digits you want to print & followed by

- f → for float
- d → for double

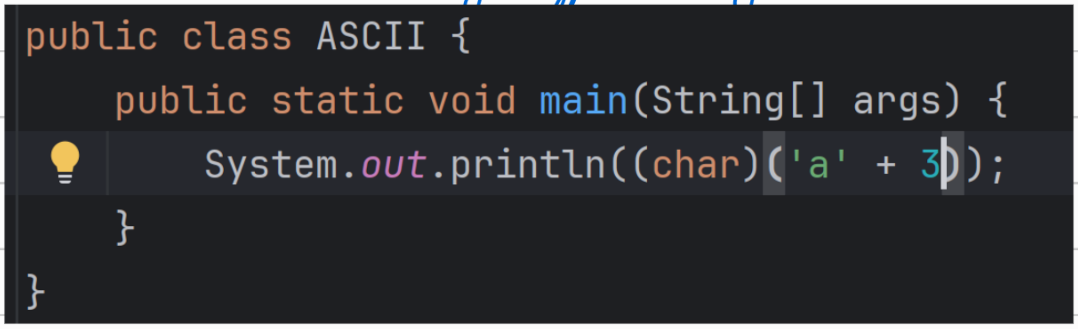
Type casting:



```java
public class ASCII {
    public static void main(String[] args) {
        System.out.println('a' + 'b');
    }
}
```

The above program would give 195 as output as it is adding the ASCII value of char.

We can fix this by type casting the entire output.



```java
public class ASCII {
    public static void main(String[] args) {
        System.out.println((char)('a' + 3));
    }
}
```

Type casting can be done as shown above.