

## Type, static type, dynamic type

In Java, every variable and expression has a type. We give examples.

- The type of a variable is given in its declaration.
- The type of a literal is determined from the literal. Examples of literals and their types appear in the box to the right.
- The type of the new-expression `new C(...)`, for any class `C`, is `C`.
- Each cast like `(int) 6.2` has the type given in the parentheses, here `int`.
- Each cast like `(Cat) c` has the type given in the parentheses, here `Cat`.
- Each method call `f(...)` of a function with a return type `t` has type `t`.
- The type of the expression `1 + 5.0` is `double`.
- The type of the expression `"1" + 5.2` is `String`.

Literal	Type
1	<code>int</code>
2.0	<code>double</code>
<code>false</code>	<code>boolean</code>
<code>null</code>	<code>Object</code>
<code>"w\$"</code>	<code>String</code>
<code>new Animal()</code>	<code>Animal</code>
<code>new Cat()</code>	<code>Cat</code>

Thus, in Java, the type of an expression is determined from its operators and the types of its operands, according to rules given in the Java language specification.

The type of an expression is a *syntactic* property; it depends only on the text form of the expression (and the rest of the program) and the operands and operators in it. It has nothing to do with execution of the program, or how an expression is evaluated at runtime.

### Static type

The *static type* of a variable is simply its type, as defined by the Java language specification. *Type* and *static type* are synonyms.

### Dynamic type

The *dynamic type* of a variable or an expression is the type of its value *during runtime*, when the program is being executed. It may change as the program is executed. For example, consider the assignment statements

```
Animal an = new Cat();  
an = new Animal();
```

where `Cat` is a subclass of `Animal`. The type, or static type, of variable `an` is `Animal`. That will never change.

But *at runtime*, after the first assignment is executed, since `an` points to an object of class `Cat`, the dynamic type of `an` is `Cat`. Execution of the second assignment statement then changes the dynamic type of `an` to `Animal`.

Thus,

- The static type of a variable is the type of the variable given in its declaration.
- The dynamic type of a variable is the type of the *value* in the variable. The term has meaning only when the program is running, and the dynamic type may be changed often as new values are assigned to the variable.

### Why use the terms static and dynamic type?

The term *static* means lacking in movement; *dynamic* means characterized by constant change.

If we were living in a world in which the only programming language was Java, we could get by without the term *static type*—and perhaps even *dynamic type*. But the terms *static type* and *dynamic type* were coined well before Java came on the scene, as people grappled with various concepts concerning types in programming languages and the terms became entrenched. We are now stuck with the terms, even if we don't need them to explain Java.