

`int sum = 0;`
declaration
variable "sum"
can only hold
integer values

optional-initialization
gives variable "sum"
an initial value

Java has 2 kinds of
variable: instance and
local.

instance variables are visible
throughout a class; every object
has a copy. Java automatically
initializes these instance variables
to 0 or its equivalent.

local variables are visible only
within a method, e.g. `sumOfSquares`.
Java does not initialize local
variables.

instance variable:

```
private int sum = 0;
```

for emphasis!
(Java would initialize anyway)

local variable:

```
int sum = 0;
```

because sum needs an initial value of 0 anyway, might as well do it here. Otherwise:

declaration → `int sum;`

initialization → `sum = 0;`

double: type of variable that can contain value with a decimal.

```
double sum = 0.00;
```

or

```
double sum = 0.;
```

boolean: type of variable
that can have one of only
two values:

true } also reserved
false } words.

void: says that a method
does not return a value
of any type to the caller.

sumOfSquares:
input n
returned (output) an int

null: reference to no object

Integer num = null;

↑
a class

↑
reference
to an
Integer
object

↑
currently
references
no object.

public: visible to software
outside the class

private: visible only inside
the class. (almost) all instance
variables are private.

Static: only one variable (or
method) for entire class,
not one per object.

final: once initialized, value
will never change.