Introducing JShell

Sander Mak
FELLOW & SOFTWARE ARCHITECT
@Sander_Mak

What Is JShell?

Read → type code

Eval → execute code

Print → see results

Loop → interactively refine

Like Lisp, Groovy, Scala

Why JShell?

Quickly test ideas

No `public static void main(String[] args)` **ceremony**

Direct feedback without compilation

Exploring APIs

Code completion

Built-in documentation

Teaching tool

No IDE, just Java

Demo

JShell Basics

Using JShell: Basics

Demo

JShell: Declarations

Using JShell: Declarations

Using JShell: Environment

```
jshell> int x = 1
x ==> 1
|  modified variable x : int
|    update overwrote variable x : int

jshell> System.out.println(
println(

jshell> System.out.println("Hello Pluralsight")
Hello Pluralsight

jshell> class Simple {
   ...>     public String simple;
   ...> }
|  created class Simple

jshell>
```

`/save mysession.jsh`

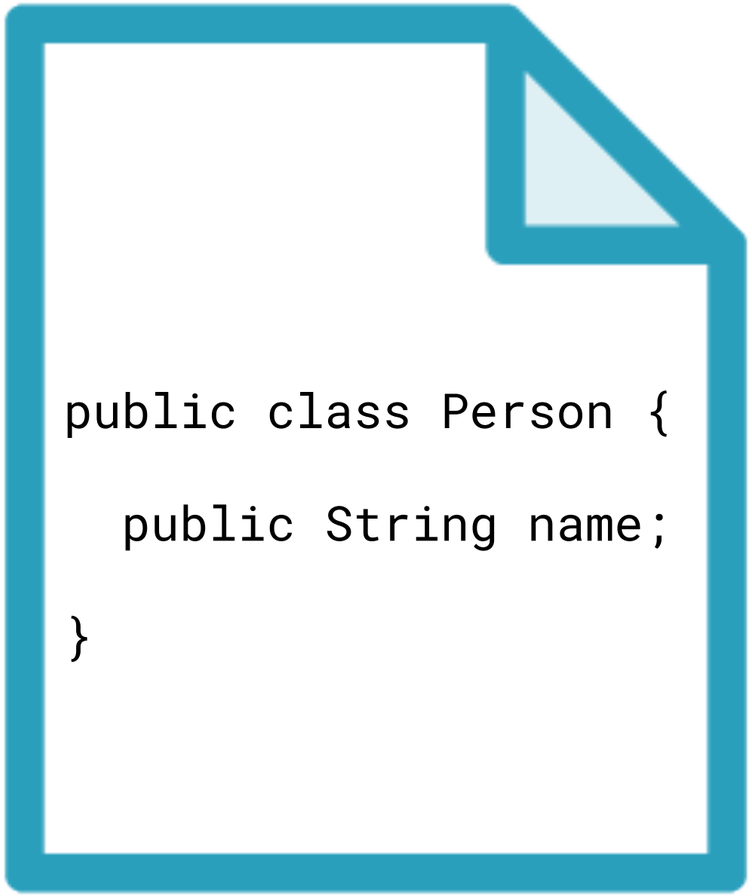
mysession.jsh

```
jshell> /open mysession.jsh
Hello Pluralsight

jshell>
```

`/open mysession.jsh`

Using JShell: Environment



Person.java

/open Person.java

```
jshell> /open Person.java

jshell> /types
|    class Person

jshell> Person p = new Person();
p ==> Person@31a5c39e

jshell> p.name = "Sander"
$3 ==> "Sander"

jshell>
```

Using JShell: Environment

Set classpath to load from JARs

```
jshell --class-path ...
```

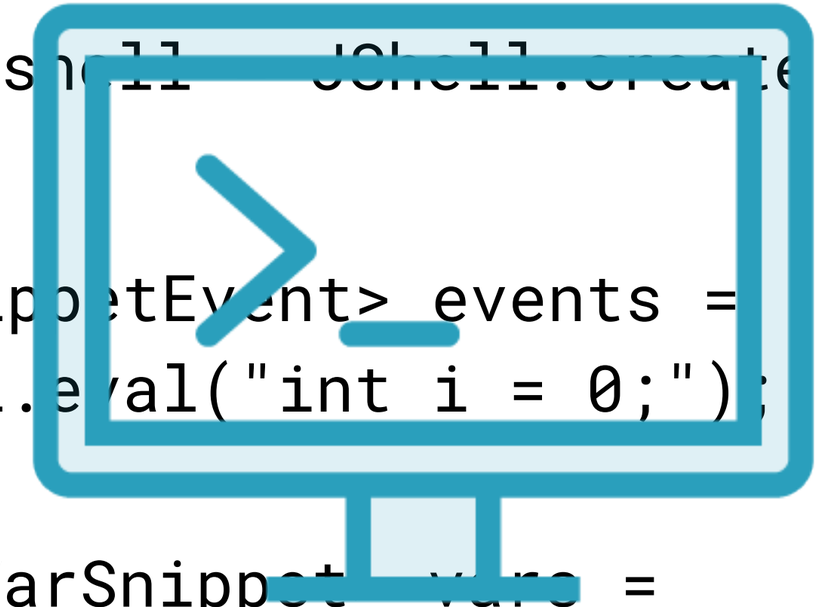
JShell API

A shell in your own application

Code-completion functionality

Source code analysis

IDE Integration



```
JShell jshell = JShell.create();

List<SnippetEvent> events =
  jshell.eval("int i = 0;");

Stream<VarSnippet> vars =
  jshell.variables();
```

Summary

Quickly test ideas

Exploring APIs

Teaching tool