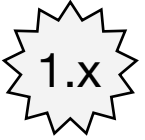


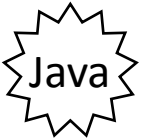
```
class JavaFXTest {  
    public static void main(String[] args) {  
        System.out.println("JavaFX"2.0");  
    }  
}
```

## Formatted Strings

```
var x = 42;  
var s = "Sense: {x}"
```



```
int x = 42;  
String s = String.format(  
    "Sense: %d", x);
```



## Sequences

```
var x = ["A", "B"], "C";  
var size = sizeof x
```



```
ObservableList<String> x =  
    FXCollections  
        .observableArrayList("A", "B", "C");  
int size = x.size();
```



## Create Range

```
var range = [3..7 step 2];  
var slice = range[0..<2]
```



```
List<Integer> range = // DIY  
List<Integer> slice =  
    range.subList(0, 2);
```



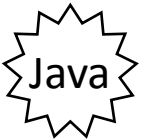
## Manipulate Sequence

```
var num = [0, 3, 8, 9, 6, 7];  
num[2..3] = [4, 5];  
num[1..0] = [1, 2]
```



```
num = ...
```

```
List<Integer> sub = num.subList(2, 4);  
sub.clear();  
sub.addAll(Arrays.asList(4, 5));  
num.addAll(1, Arrays.asList(1, 2));
```

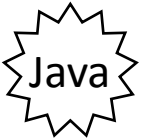


## Sequence Select

```
var num = [0, 3, 8, 9, 6, 7];  
var e = num[x | x mod 2 == 0]
```



```
num = ...  
e = ...  
for (int x : num) {  
    if (x % 2 == 0) {  
        e.add(x);  
    }  
}
```



# Trigger

```
var num = [] on replace old[lo..hi] = x {  
    println("lo: {lo}, hi: {hi}");  
    println("old: {old}, x: {x}");  
    println("result: {num}");  
}
```



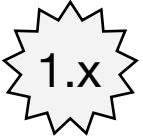
```
ObservableList<Integer> num =  
    FXCollections.observableArrayList();  
num.addListener(  
    new ListChangeListener<Integer>() {  
        public void onChanged(Change<? extends Integer> c) {  
            while (c.next()) {  
                System.out.printf(  
                    "lo: %d, hi: %d%n" +  
                    "old: %s, new: %s%n" +  
                    "result: %s%n",  
                    c.getFrom(), c.getTo(),  
                    c.getRemoved(), c.getAddedSubList(),  
                    c.getList());  
            } } } );
```



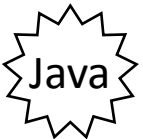
see [javafx.collections.\\*](#)

# Functions

```
function mul(a : Number, b : Number) : Number {  
    return a * b;  
}  
function foo(bar: function(Number, Number) : Number) :  
    function(Number): Number {  
    return function(x : Number) { return bar(x, x); }  
}  
println(foo(mul) (3))
```



```
interface OneParamFunc { double op(double x); }  
interface TwoParamFunc { double op(double a, double b); }  
static TwoParamFunc mul = new TwoParamFunc() {  
    public double op(double a, double b) { return a * b; }  
};  
static OneParamFunc foo(final TwoParamFunc bar) {  
    return new OneParamFunc() {  
        public double op(double x) { return bar.op(x, x); }  
    };  
}  
System.out.println(foo(mul).op(3));
```





# Binding

```
var x : Integer;  
var y : Integer = bind x with inverse;  
println("x: {x}, y: {y}");  
  
x = 1;  
println("x: {x}, y: {y}");  
  
y = 2;  
println("x: {x}, y: {y}")
```



```
IntegerProperty x = new SimpleIntegerProperty();  
IntegerProperty y = new SimpleIntegerProperty();  
y.bindBidirectional(x);
```



```
System.out.printf("x: %d, y: %d%n", x.get(), y.get());  
  
x.set(1);  
System.out.printf("x: %d, y: %d%n", x.get(), y.get());  
  
y.set(2);  
System.out.printf("x: %d, y: %d%n", x.get(), y.get());
```

see [javafx.beans.property.\\*](#)

# User Interface

```
Stage {  
  title: "Hello Application", width: 250, height: 80  
  scene: Scene {  
    content: Text { x: 10, y: 30, content: "Hello" }  
  }  
}
```



```
class Hello extends Application {  
  public static void main(String[] args) {  
    Application.launch(args);  
  }  
  @Override public void start(Stage primaryStage) {  
    primaryStage.setTitle("Hello Application");  
    primaryStage.setWidth(250);  
    primaryStage.setHeight(80);  
    Group root = new Group();  
    Scene scene = new Scene(root);  
    Text text = new Text(10, 30, "Hello");  
    root.getChildren().add(text);  
    primaryStage.setScene(scene);  
    primaryStage.show();  
  } }  
see javafx.application.*, javafx.stage.*, javafx.scene.*
```



# User Interface Builder



```
Stage stage = new Stage();
stage.setTitle("Hello Application");
stage.setWidth(250);
stage.setHeight(80);
Group root = new Group();
Scene scene = new Scene(root);
Text text = new Text(10, 30, "Hello");
root.getChildren().add(text);
stage.setScene(scene);
```

```
Stage stage = StageBuilder.create()
    .title("Hello Application")
    .width(250)
    .height(80)
    .scene(
        SceneBuilder.create().root(
            GroupBuilder.create().children(
                TextBuilder.create().x(10).y(30).text("Hello").build()
            ).build()
        ).build()
    ).build();
```

see `javafx.application.*`, `javafx.stage.*`, `javafx.scene.*`