This is cucumber expression what we have written:

```
20    @When("I search for a product with name {string} and price {int}")
```

We cannot mix regular expression and cucumber expression in a single method-

```
20    @When("I search for a product with name {string} and price (\\d+)")
21    public void i_search_for_a_product_with_name_and_price(String productName, Integer price) {
22        System.out.println("Step 2: Search the product with name : " + productName + " price: " + price
23        product = new Product(productName, price);
24    }
25
26    @Then("Product with name {string} should be displayed")
```

Else we get error. And step definition will also be flagged.

```
22
23 Scenario: Search a Product
24 Given I have a search field on Amazon Page
25 When I search for a product with name "Apple MacBook Pro" and price 1000
26 Then Product with name "Apple MacBook Pro" should be displayed
27 Then Order id is 12345 and username is "Naveen"
28
```

See line 20 where we have used both forms of expression.

But inside a file, for example, if there are ten methods, two of them can have regular expression only and 8 of them can have cucumber expression only.

Regular expression-

Used to fetch certain values.

Quantifiers in regular expression-

How many times a character needs to occur.

Capture group –

The “()” brackets are known as capture group.

```
1 Two types of REg Expressions in Cucumber:
2
3 1. Regular Expression --> [0-9]+, (\\d+)
4 2. Cucumber Expression (2017)
5
6 rules:
7 1. Step def file will be generating cucumber exp by default
8 2. But you can use regular exp also in step def file
9 3. You can mix both regular and cucumber exp in step definition file
10 4. but you can not mix both expressions in a step definition method
11
12 {string} {int} {float}-- cucumber expression
13
14 Regular Expressions:
15
16 ([0-9]) ->capture group --> 0 to 9 digits appear
17
```

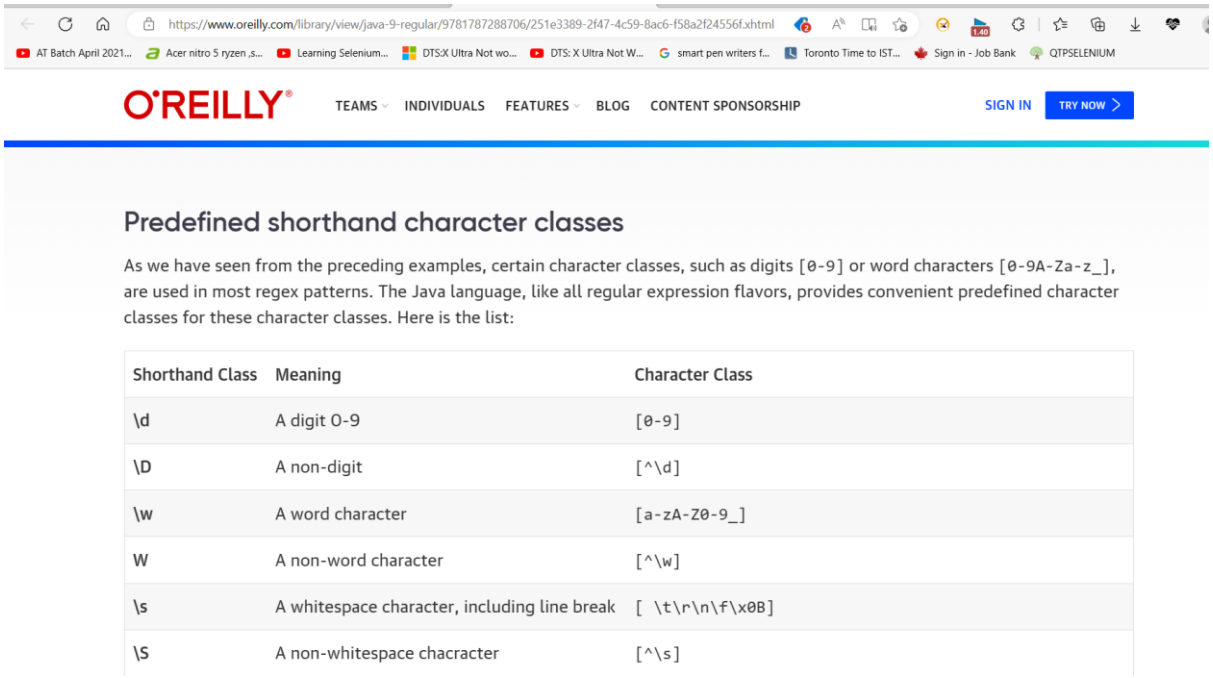
Note-

{n} says total number of occurrences of the letter or digit.

```
18 Quantifiers in Reg Exp: + * ? {n}
```

```
18 Quantifiers in Reg Exp: + * ? {n}
19 Define -> How many times a character needs to be occurred
20
21 ([0-9]+) -> 0 to 9 digits appear  (once or more)
22 ([0-9]{4}) --> 0000 , 9999, 1212, 3456, 1234, 8888
23 ([0-9]*) -> zero or more
24 ([0-9]?) -> zero or once
25
```

Shorthand characters guide-

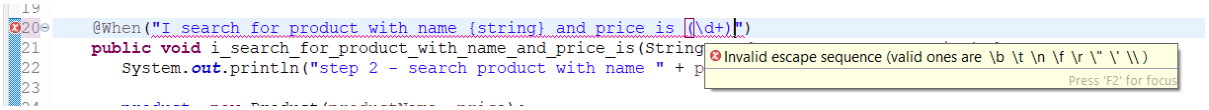


[Predefined shorthand character classes - Java 9 Regular Expressions [Book] (oreilly.com)]

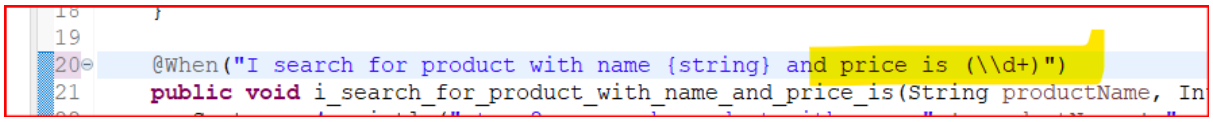
```
20 https://www.oreilly.com/library/view/java-9-regular/9781787288706/251e3389-2f47-4c59-8ac6-f58a2f24556f.xhtr
```

```
Short hand characters:

\d -- numeric digits
(\d+)

([a-zA-Z0-9]+)
```

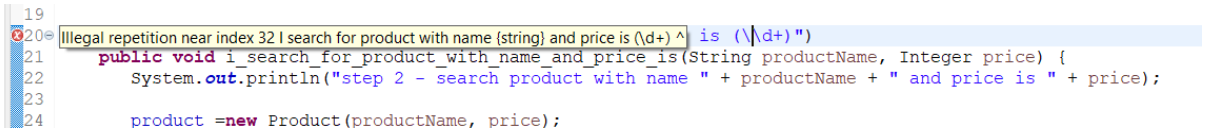
Invalid escape sequence error displayed when we use regular expression-



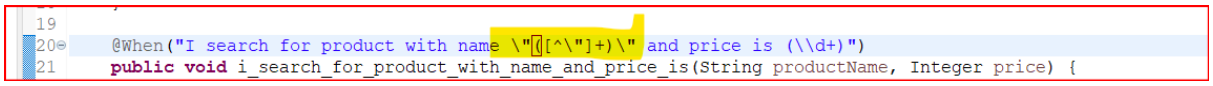
This is because we need to one more slash before "\". See in the above error, the list of valid escape sequences.



Still, we will get error for mixing regular expression with cucumber expression:



This is regular expression and it will capture all the strings except double quote:



The highlighted step is not flagged, which means our regular expression is correct.

```
Given I have a search field on Amazon page
#given can be considered as pre-condition
When I search for product with name "apple" and price is 1000
#string is in double quotes
```

Run the search feature file and we should get error:

```
io.cucumber.core.exception.CucumberException: Step [I search for product
with name "([^"]+)" and price is (\d+)] is defined with 2 parameters at
'StepDefinitions.SearchFeatureStepDef.i_search_for_product_with_name_and_
price_is(java.lang.String,java.lang.Integer)'.
However, the gherkin step has 0 arguments.
Step text: I search for product with name "apple" and price is 1000
      at
io.cucumber.core.runner.PickleStepDefinitionMatch.arityMismatch(PickleSte
pDefinitionMatch.java:83)
      at
io.cucumber.core.runner.PickleStepDefinitionMatch.runStep(PickleStepDefin
itionMatch.java:42)
      at
io.cucumber.core.runner.ExecutionMode$1.execute(ExecutionMode.java:10)
      at io.cucumber.core.runner.TestStep.executeStep(TestStep.java:92)
      at io.cucumber.core.runner.TestStep.run(TestStep.java:64)
```

```
	at
io.cucumber.core.runner.PickleStepTestStep.run(PickleStepTestStep.java:51
)
	at io.cucumber.core.runner.TestCase.run(TestCase.java:104)
	at io.cucumber.core.runner.Runner.runPickle(Runner.java:71)
	at
io.cucumber.core.runtime.Runtime.lambda$execute$5(Runtime.java:110)
	at
io.cucumber.core.runtime.CucumberExecutionContext.runTestCase(CucumberExe
cutionContext.java:117)
	at
io.cucumber.core.runtime.Runtime.lambda$execute$6(Runtime.java:110)
	at
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
	at java.util.concurrent.FutureTask.run(FutureTask.java:266)
	at
io.cucumber.core.runtime.Runtime$SameThreadExecutorService.execute(Runtim
e.java:233)
	at
java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorServi
ce.java:112)
	at io.cucumber.core.runtime.Runtime.lambda$run$2(Runtime.java:86)
	at
java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)
	at java.util.stream.SliceOps$1$1.accept(SliceOps.java:204)
	at
java.util.ArrayList$ArrayListSpliterator.tryAdvance(ArrayList.java:1359)
	at
java.util.stream.ReferencePipeline.forEachWithCancel(ReferencePipeline.ja
va:126)
	at
java.util.stream.AbstractPipeline.copyIntoWithCancel(AbstractPipeline.jav
a:498)
	at
java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:485)
	at
java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:4
71)
	at
java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708
)
	at
java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
	at
java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:499)
	at io.cucumber.core.runtime.Runtime.run(Runtime.java:87)
	at io.cucumber.core.cli.Main.run(Main.java:92)
	at cucumber.api.cli.Main.run(Main.java:30)
	at cucumber.api.cli.Main.main(Main.java:15)
```

How to resolve the above error:

Open the step definition file of say, search feature.

Add “^” at start and “$” at end.

Then run the search feature file.

```
19
20⊖    @When("^I search for product with name \"([^\"]+)\" and price is (\\d+)$")
21     public void i_search_for_product_with_name_and_price_is(String productName, Int
```

Output:

```
Jun 12, 2023 4:51:27 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use
io.cucumber.core.cli.Main

Scenario: Search for a product                                    #
src/test/resources/AppFeatures/Search.feature:24
step 1 - i am on search page
  Given I have a search field on Amazon page                     #
StepDefinitions.SearchFeatureStepDef.i_have_a_search_field_on_amazon_page
()
step 2 - search product with name apple and price is 1000
  When I search for product with name "apple" and price is 1000 #
StepDefinitions.SearchFeatureStepDef.i_search_for_product_with_name_and_p
rice_is(java.lang.String,java.lang.Integer)
step 3 - product with apple : is displayed
returned product is apple
  Then Product with name "apple" should be displayed             #
StepDefinitions.SearchFeatureStepDef.product_with_name_should_be_displaye
d(java.lang.String)

1 Scenarios (1 passed)
3 Steps (3 passed)
0m0.371s
```

Codes-

Search feature file:

```
#Author: your.email@your.domain.com
#Keywords Summary :
#Feature: List of scenarios.
#Scenario: Business rule through list of steps with arguments.
#Given: Some precondition step
#When: Some key actions
#Then: To observe outcomes or validation
#And,But: To enumerate more Given,When,Then steps
#Scenario Outline: List of steps for data-driven as an Examples and
<placeholder>
#Examples: Container for s table
#Background: List of steps run before each of the scenarios
#""" (Doc Strings)
#| (Data Tables)
#@ (Tags/Labels):To group Scenarios
#<> (placeholder)
#"""
## (Comments)
#Sample Feature Definition Template


Feature: Amazon Search
#this is the feature for which we want to write code and requirements
#one feature file can have multiple scenarios
Scenario: Search for a product
```

```
#here we give the scenario name
#below scenario (given, when, then, and) etc are known as steps
Given I have a search field on Amazon page
#given can be considered as pre-condition
When I search for product with name "apple" and price is 1000
#string is in double quotes
Then Product with name "apple" should be displayed
```

Step definition file for search feature:

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import junit.framework.Assert;
import amazonImplementation.Product;
import amazonImplementation.Search;

public class SearchFeatureStepDef {

	Product product;
	Search search;

	@Given("I have a search field on Amazon page")
	public void i_have_a_search_field_on_amazon_page() {
		System.out.println("step 1 - i am on search page");
	}

	@When("^I search for product with name \"([^\"]+)\" and price is (\\d+)$")
	public void i_search_for_product_with_name_and_price_is(String productName, Integer
price) {
		System.out.println("step 2 - search product with name " + productName + " and
price is " + price);

		product = new Product(productName, price);
	}

	@Then("Product with name {string} should be displayed")
	public void product_with_name_should_be_displayed(String productName) {
		System.out.println("step 3 - product with " + productName + " : is displayed");
		search = new Search();
		String productNameReturned = search.displayProductName(product);
		System.out.println("returned product is " + productNameReturned);
		Assert.assertEquals(product.getProductName(), productNameReturned);
	}

}
```

Output:

```
Jun 12, 2023 4:51:27 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use
io.cucumber.core.cli.Main
```

```
Scenario: Search for a product                                          #
src/test/resources/AppFeatures/Search.feature:24
step 1 - i am on search page
  Given I have a search field on Amazon page                            #
StepDefinitions.SearchFeatureStepDef.i_have_a_search_field_on_amazon_page
()
step 2 - search product with name apple and price is 1000
  When I search for product with name "apple" and price is 1000 #
StepDefinitions.SearchFeatureStepDef.i_search_for_product_with_name_and_p
rice_is(java.lang.String,java.lang.Integer)
step 3 - product with apple : is displayed
returned product is apple
  Then Product with name "apple" should be displayed                    #
StepDefinitions.SearchFeatureStepDef.product_with_name_should_be_displaye
d(java.lang.String)

1 Scenarios (1 passed)
3 Steps (3 passed)
0m0.371s
```