

Japanese Tutorial 4 - Regular expressions basics

==~Dan Qualye'potatoe' Dan Qualye [198993](#) potatoe potato

```
"potatoe" ==~ /potatoe/      ==> true
```

groovy true 1==~==equality 2 // groovy
potato 'e'?' true

```
"potatoe" ==~ /potatoe?/    ==> true
```

```
"potato" ==~ /potatoe?/    ==> true
```

```
"motato" ==~ /potatoe?/    ==> false
```

boolean

Pete Wisniewski Wisniewski (Pete Wisniewski)

```
def checkSpelling(spellingAttempt, spellingRegularExpression)
{
  if (spellingAttempt ==~ spellingRegularExpression)
  {
    println("Congratulations, you spelled it correctly.")
  }
  else
  {
    println("Sorry, try again.")
  }
}
theRegularExpression = /Wisniewski/
checkSpelling("Wisniewski", theRegularExpression)
checkSpelling("Wisnewski", theRegularExpression)
```

(function) (anonymous)

if ==~

'w'

```
theRegularExpression = /Wisniew?ski/
checkSpelling("Wisniewski", theRegularExpression)
checkSpelling("Wisnieski", theRegularExpression)
checkSpelling("Wisniewewski", theRegularExpression)
```

? 'w' (spellingAttempt'Wisniewski' 'Wisnieski' groovycheckSpelling

'?'

'ie'

```

theRegularExpression = /Wisn(ie|ei)w?ski/
checkSpelling("Wisniewski", theRegularExpression)
checkSpelling("Wisneiwski", theRegularExpression)
checkSpelling("Wisnieski", theRegularExpression)
checkSpelling("Wisneiski", theRegularExpression)

checkSpelling("Wisniewski", theRegularExpression)
checkSpelling("Wisneiwski", theRegularExpression)

```

"Wisniewski", "Wisneiwski", "Wisnieski", "Wisneiski" | "ie" "ei" ()
 ([) Peter

```

theRegularExpression = /Wis\[abcd\]niewski/ // 'a', 'b', 'c', 'd'
theRegularExpression = /Wis\[abcd\]?niewski/ // 'a', 'b', 'c', 'd',
theRegularExpression = /Wis\[a-zA-Z\]niewski/ //
theRegularExpression = /Wis\[^\abcd\]niewski/ // 'a', 'b', 'c', 'd'

```

[) ^

a?	0 1 *a*	'a'
a*	0 *a*	'a', 'aa', 'aaa'
a+	1 1 *a*	'a', 'aa', 'aaa'
a b	*a* *b*	'a' 'b'
.	1	'a', 'q', 'l', '_', '+'
[woeirjds]		'w', 'o', 'e', 'i', 'r', 'j', 's', 'd'
[1-9]		'1', '2', '3', '4', '5', '6', '7', '8', '9'
[^13579]		
(ie)	('ie'
^a	*a*	'a'
a\$	*a*	'a'

\'

```

// evaluates to true, and will for anything ending in a question mark (that doesn't
have a question mark in it)
"How tall is Angelina Jolie?" =~ /^[^?]+\?/

```

PERLPERL

google[she is|http://www.google.com/search?hl=en&q=how+tall+is+angelina+jolie&btnG=Google+Search]

/	[^?]	+	?	/
	'?'	1		

? \