

# Regular Expressions Cheat Sheet

## Anchors

<b>^</b>	Anfang des Strings oder der Zeile
<b>\A</b>	Anfang des Strings oder der Zeile
<b>\$</b>	Ende des Strings oder der Zeile
<b>\Z</b>	Ende des Strings oder der Zeile
<b>\b</b>	Wortgrenze
<b>\B</b>	Keine Wortgrenze
<b>&lt;</b>	Anfang des Wortes
<b>&gt;</b>	Ende des Wortes

## Special Sequences

<b>.</b>	Jedes Zeichen, bis auf neue Zeile \n
<b>\s</b>	Weißraum (Leerzeichen, Tabs)
<b>\S</b>	Kein Weißraum
<b>\d</b>	Zahlen
<b>\D</b>	Keine Zahlen
<b>\w</b>	Worte (Buchstaben und Zahlen)
<b>\W</b>	Keine Worte

## Quantifiers

<b>*</b>	0 oder mehr
<b>+</b>	1 oder mehr
<b>?</b>	0 oder 1
<b>{3}</b>	Genau 3
<b>{3,}</b>	3 oder mehr
<b>{3,5}</b>	3, 4 oder 5
Mit "?" Non-Greedy.	

## Assertions

<b>?=</b>	Lookahead assertion
<b>?!</b>	Negative lookahead
<b>?&lt;=</b>	Lookbehind assertion
<b>?!= or ?&lt;!</b>	Negative lookbehind
<b>?&gt;</b>	Once-only Subexp-ression
<b>?()</b>	Condition [if then]
<b>?() </b>	Condition [if then else]
<b>?#</b>	Kommentar

## Sets, Ranges & Groups

<b>[abc]</b>	Set beinhaltet (a or b or c)
<b>[^abc]</b>	Set beinhaltet nicht (a or b or c)
<b>a-z</b>	Range - Kleinbuchstaben von a bis z
<b>A-Z</b>	Range - Großbuchstaben von A bis Z
<b>0-9</b>	Range Zahlen 0 bis 9
<b>(...)</b>	(Capturing) Gruppe
<b>(a b)</b>	a oder b
<b>(?....)</b>	Passive (non-capturing) Gruppe

## Special Characters

<b>^</b>	<b>[</b>	<b>.</b>
<b>{</b>	<b>*</b>	<b>(</b>
<b>+</b>	<b>)</b>	<b> </b>
<b>&lt;</b>	<b>&gt;</b>	<b>\$</b>
<b>\</b>	<b>?</b>	

## Escape Sequences

<b>\</b>	Escape "Special Characters"
----------	-----------------------------

## Python Main Functions

### re.match(r'pattern',string,flags=0)

Rückgabewert:	MatchObject
---------------	-------------

### re.fullmatch(r'pattern',string,flags=0)

Rückgabewert:	MatchObject
---------------	-------------

### re.search(r'pattern',string,flags=0)

Rückgabewert:	MatchObject
---------------	-------------

### re.sub(r'pattern',repl, string, count=0, flags=0)

Rückgabewert:	String
---------------	--------

### re.subn(r'pattern',repl, string, count=0, flags=0)

Rückgabewert:	Tuple
---------------	-------

### re.findall(r'pattern', string, flags=0)

Rückgabewert:	Liste
---------------	-------

### re.finditer(r'pattern', string, flags=0)

Rückgabewert:	Iterator
---------------	----------

### re.split(r'pattern', string, maxsplit=0, flags=0)

Rückgabewert:	Liste
---------------	-------

## Python Sub Functions

### re.compile(pattern)

Rückgabewert:	RegexObject
---------------	-------------

### match.group()

Rückgabewert:	String
---------------	--------

## String Replacements

<b>\n</b>	n-te nicht-passive Gruppe
<b>\2</b>	"-xyz-" in ^(abc-(xy-z))\$
<b>\1</b>	"-xyz-" in ^(?:a-bc)-(xyz)\$