

# Лекция 12

## Регулярни изрази

### част 2



## Квантификатори

- \* - нула или повече пъти.  
/ab\*c/ => 'ac', 'abc', 'abbc', 'abbbc' и т.н., но не и 'abxc'
- + - един или повече пъти  
/ab+c/ => 'abc', 'abbc', 'abbbc' и т.н., но не и 'ac' или 'abxc'
- {d} , където d е произволна цифра  
/ab{3}c/ => 'abbbc', 'babbbc' и пр., но не и 'abc' или 'abbbc'
- {d1, d2} , където d1, d2 са произволни цифри  
/ab{1, 3}c/ => 'abc', 'abbbc' и пр., но не и "ac" или "abbbc"

## Групиране и избор

- **()** - групира низа в скобите

/ala bala+ / => 'ala bala', 'ala balaaaa'

/ala (bala)+ / => 'ala **bala**', 'ala **bala bala**',...

но не и 'ala **bala**aaa'

- **|** - за избор на един от няколко елемента.

/iv(o|a|eli)na/ => "ivona", "ivana", "ivelina".

/b(u|a)m/ => "bum", "bam", но не и "buam", ...

## Класове от символи

- **[ ... ]** - съпоставя се със един символ от зададеното множество от символи. Ако имаме: `[sdt]` => ще се съпостави с 's' или 'd' или 't'. Могат да се указват интервали. Например:

`[1-5]` <=> `[1|2|3|4|5]`

- Съкратени записи за някои по често използвани класове:

`\d` - цифра ( еквивалентно на: `[0-9]` )

`\w` - буквено-цифров символ ( всички букви, цифри и символа ' \_ ' ).

`\s` - бял спейс (интервал, табулация, нов ред, ...)

И съответните им отрицания: `\D`, `\W`, `\S`

## Автоматични променливи при РИ

- `$&` – съдържа онази част от низа, която **съвпада** с регулярния израз.
- `$`` – онази част от низа, която **предхожда** частта съвпаднала с регулярния израз.
- `$'` – онази част от низа, която **следва** частта, съвпаднала с регулярния израз.

```
$str="bimboombam";  
if ($str=~bo+m/) {  
    print "$` - $& - $'";  
}
```

ще изведе: bim - boom - bam

## Запаметяване при РИ

- **\$1, \$2, ...** - в тези променливи се съхраняват онези части от низа, който са съвпаднали със съответните части от шаблона, заградени в скоби - (). Например

```
$str='i.e.popova@gmail.com';  
if ($str =~ /(.+)@(\w+)\.(\.+)/ ){  
    print "$1 [at] $2 [dot] $3"  
}
```

ще изведе на екрана: `i.e.popova [at] gmail [dot] com`

Тези променливи се използват често при оператора за заместване:

```
$str="pisateliat Ivan Vazov ...";  
if ($str =~ s/([A-Z])\w+\s+([A-Z]\w+)/$1.$2/ ){  
    print "$str"  
}
```

ще изведе: `pisateliat I.Vazov ...`

## Модификатори

- Модифицират работата на операторите за съпоставяне. Поставят се след самия оператор, например: `m/regex/i`

- Най-често използвани са:

- **i** - съпоставяне без отчитане на малки/големи букви:

`/iva/i` - съпоставя се както с 'iva' така и с 'Iva', ..., 'IVA'

- **g** - съпоставянето се извършва глобално. Т.е. работата на оператора не спира при първото намерено съвпадение, а продължава до изчерпването на всички такива:

```
$str="Georgi Arnaudov and Anzhela Tosheva";  
while ($str =~ s/([A-Z])\w+\s+([A-Z]\w+)/$1.$2/g ) {  
    print "$str"  
}
```

ще изведе: G.Arnaudov and A.Tosheva