# COMP233-04B Internet Applications

Lecture 19:
Regular Expressions

## Matching Substrings

Expression

```
mall
```

matches the string

```
smallest
```

because it contains a matching string.
To avoid this, use ^ and $:

```
^mall$
```

## Regular Expression Characters

| | |
|---|---|
| a | Character matches itself |
| . | Match any character |
| [a-z] | Match any of these characters |
| [^ABC] | Match any character except these |
| ^ | Match to start of string |
| $ | Match to end of string |

## Whitespace

- Users often enter whitespace at the beginning and end of their input.
- We generally want to ignore this.

```
^ *small *$
```

## Examples

```
[mM]ule
```

matches:

```
mule, Mule
```

```
[mM].le
```

matches:

```
male, Mole, mule, MMle, …
```

## Another Example

- Allow the user to type a sentence with a two-digit number (10-99) in it somewhere.

```
[1-9][0-9]
```

- How can we find out which number was found in the input?
- JavaScript returns it as result!

## Result of the `match()` Method

```
var text = document.myform.mytext.value;
var regstr = document.myform.regexp.value;
var regexp = new RegExp(regstr);
var result = text.match(regexp);
if (result) {
  // matches!
  document.bgColor = "lightgreen";
  document.myform.matched.value =
      result[0];
}
```

## More Characters

| | |
|---|---|
| \d | A digit (0…9) |
| \D | Anything but a digit |
| \w | A word character (letter, digit, …) |
| \W | Anything but a word character |
| \s | Whitespace (space or tab) |
| \S | Anything but whitespace |

## Combining Regular Expressions

| | |
|---|---|
| (…) | Grouping |
| * | Zero or more times |
| + | One or more times |
| ? | Zero or one times (optional) |
| {5} | Exactly 5 times |
| {3,6} | 3–6 times |

Applied after subepxressions, e.g., `[aA]*`.

## Escaping Special Characters

**Problem:**

How to match strings containing special characters like

`. ^ $ [ ] \`

**Solution:**

Escape them with `\`

`[a-z]+\.txt`

## More Examples

Any word of letters:

`[a-zA-Z]*`

Any nonempty word of letters:

`[a-zA-Z]+`

Any nonempty word of letters that starts with a capital:

`[A-Z][a-z]*`

## Final Note

Regular Expressions can be combined with OR,

e.g.,

```
large|medium|small
  [a-z]*|[0-9]*
```

But …

there is no AND.