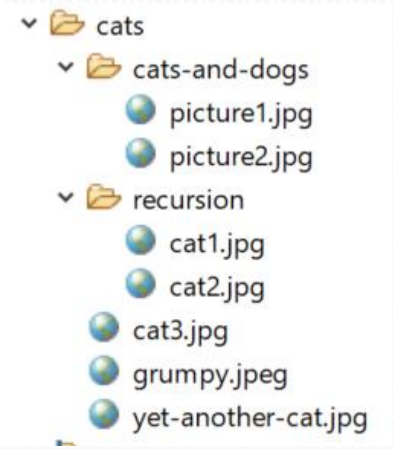
Plan for Lecture

1. Review code
2. Fix style and add indentation to output
3. Grammars and Regular Expressions

print



```
cats
  cats-and-dogs
    picture1.jpg
    picture2.jpg
  recursion
    cat1.jpg
    cat2.jpg
  cat3.jpg
  grumpy.jpeg
  yet-another-cat.jpg
```

```
public static void print(File file) {

}
```

```
public static void print(File file) {
    if (!file.isDirectory()) {
        System.out.println(file.getName());
    } else {
        System.out.println(file.getName());
        File[] subFiles = file.listFiles();
        for (int i = 0; i < subFiles.length; i++) {
            print(subFiles[i]);
        }
    }
}
```

file = recursion

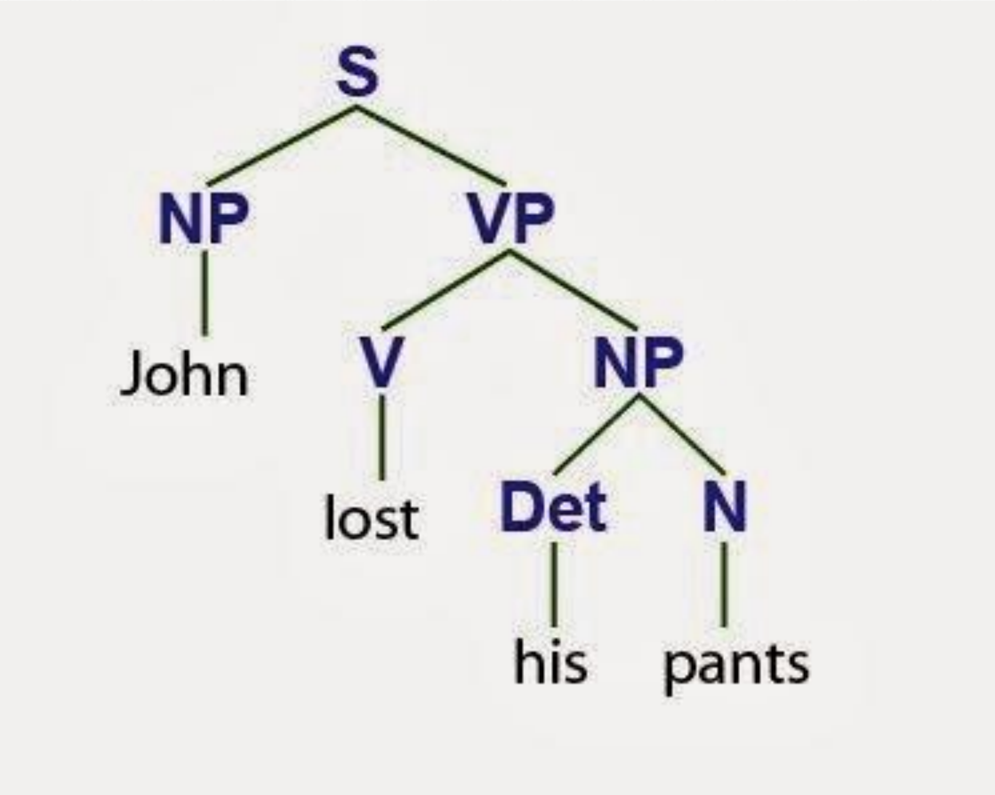
…

```
) {

}
```

file = picture2.jpg

```
cats
cats-and-dogs
picture1.jpg
picture2.jpg
recursion
…
```


S
NP
VP
John
V
NP
lost
Det
N
his
pants

Languages and grammars

- (formal) **language**: A set of words or symbols.

- **grammar**: A description of a language that describes which sequences of symbols are allowed in that language.
  - describes language *syntax* (rules) but not *semantics* (meaning)
  - can be used to generate strings from a language, or to determine whether a given string belongs to a given language

Backus-Naur (BNF)

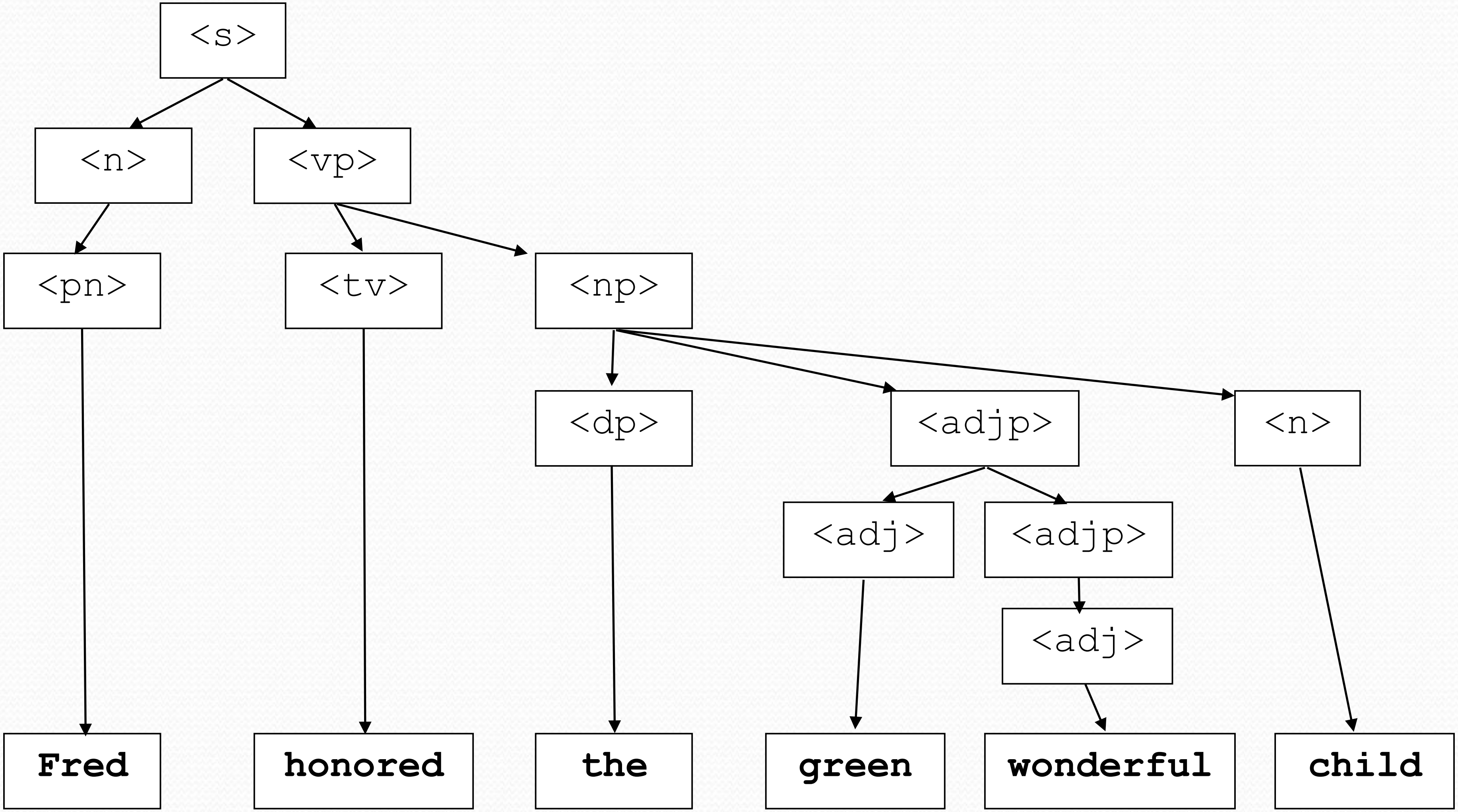
- **Backus-Naur Form (BNF)**: A syntax for describing language grammars in terms of transformation *rules*, of the form:

  <**symbol**> ::= <**expression**> | <**expression**> ... | <**expression**>

  - **terminal**: A fundamental symbol of the language.
  - **non-terminal**: A high-level symbol describing language syntax, which can be transformed into other non-terminal or terminal symbol(s) based on the rules of the grammar.

  - developed by two Turing-award-winning computer scientists in 1960 to describe their new ALGOL programming language

Sentence generation


<s>
<n>
<vp>
<pn>
<tv>
<np>
<dp>
<adjp>
<n>
<adj>
<adjp>
<adj>
Fred
honored
the
green
wonderful
child