

# Coding Syllabus

---

## GRADES 5 - 8: JUNIOR PYTHON CODING

### 1. Basic Syntax

- Assignment and rebinding
- Indentation rules
- `print()` usage with end argument
- Tuple returns and unpacking

### 2. Arguments

- Positional vs keyword arguments
- Default parameter values
- Keyword argument calls and order
- Argument unpacking basics

### 3. Strings

- Indexing and slicing (`s[1:4]`, `s[::-1]`)
- String methods: `split()`, `strip()`, `upper()`, `count()`
- Difference between `"` and `'''` in length and equality

### 4. While Loops

- Loop control and counters
- Natural stopping conditions via decrement
- `break` behavior and `while ... else` usage

### 5. Variables

- Rebinding vs aliasing
- Immutable types (`int`, `str`)
- Function calls not changing immutables

### 6. If Statements

- Proper `if` / `elif` / `else` structure
- Logical expressions (`x and (x > 5)`)
- Condition ordering and truth evaluation

### 7. Functions

- Definition and calling
- Return values (including tuples)
- Pure vs side-effect functions

### 8. Parameters

- Parameter lists with default values
- Keyword-only and positional argument rules
- `*args` handling and counting
- Invalid parameter mixes causing errors

# Coding Syllabus

---

## GRADES 5 - 8: JUNIOR PYTHON CODING

### **9. Advanced Strings**

- Reversing and slicing combinations (`s[::-1][1:4]`)
  - Effects of delimiters in `split()`
-

# Coding Syllabus

---

## GRADES 9 - 12: SENIOR PYTHON CODING

### 1. Basic Syntax

- f-strings; expression-in-f-string conditionals
- Operator/parentheses awareness (used across questions)

### 2. Arguments

- Positional vs keyword
- Default values (mutable-default pitfall)
- Unpacking with `*` and `**`

### 3. Strings

- Slicing incl. negatives/steps (e.g., `s[-2:2:-2]`)
- `strip()`, `split()`, `join()`
- Immutability and rebinding

### 4. Advanced While Loops

- Loop conditions, precision caveats (float increments)
- `while ... else` behavior
- `continue/break` interaction

### 5. Variables

- Rebinding vs mutation (aliases)
- Scope: local, `global`, `nonlocal`

### 6. If Statements

- `if / elif / else`
- Chained comparisons (`10 <= x < 20`)
- Ternary/conditional expressions

### 7. Arithmetic

- `+` `-` `*` `/` `//` `%` `**` and precedence
- Bitwise ops used: `|`, shifts `<<`

### 8. Counters

- Accumulators in loops
- Nested-loop counting patterns

### 9. Boolean Logic

- `and`, `or`, `not` precedence
- Short-circuit evaluation (with function side-effects)

### 10. Input Handling

- `input()` → type conversion (`int`)

# Coding Syllabus

---

## GRADES 9 - 12: SENIOR PYTHON CODING

- Validation + raising errors
- `try/except/finally` flow

### **11. Break and Continue Statements**

- Effect inside `for/while`
- Impact on `for...else` / `while...else`

### **12. Parameters**

- `*args, **kwargs`
- Keyword-only params (`c=`)
- Binding/conflict rules

### **13. Functions**

- Definitions and calls
  - Closures with `nonlocal`
  - Lambdas in pipelines
  - Simple recursion (base vs recursive case)
-