Math as Code Cheatsheet

Introduction

This cheatsheet provides a bridge between formal mathematical notation and how those concepts are typically implemented in programming languages (C++, Python, JS, etc.).

Basic Arithmetic & Algebra

Math Notation	Code Implementation (e.g. Python)
$x + y$	`x + y`
$x - y$	`x - y`
$x \cdot y, xy$	`x * y`
$\frac{x}{y}$	`x / y`
$x^n, x \times x \times \dots$	`x ** n` or `pow(x, n)`
$\sqrt{x}$	`math.sqrt(x)` or `x ** 0.5`
$\lvert x \rvert$	`abs(x)`

Sets and Logic

Math Notation	Code Implementation
$x \in A$	`x in A` (e.g. searching in list/set)
$A \cup B$	`A | B` or `A.union(B)`
$A \cap B$	`A & B` or `A.intersection(B)`
$P \wedge Q$	`P and Q` or `P && Q`
$P \vee Q$	`P or Q` or `P || Q`
$\neg P$	`not P` or `!P`
$\forall x \in A, f(x)$	`all(f(x) for x in A)`
$\exists x \in A, f(x)$	`any(f(x) for x in A)`

Summation and Product

Math Notation	Code Implementation
$\sum_{i=0}^{n} i$	`sum(range(n + 1))` or a `for` loop
$\prod_{i=1}^{n} i$	`math.prod(range(1, n + 1))` or `factorial(n)`
$n!$	`math.factorial(n)`

Functions & Calculus

- **Function Definition**:
  - Math: $f(x) = x^2$
  - Code: `def f(x): return x ** 2`
- **Piecewise Function**:
  - Math: $f(x) = \begin{cases} x \text{ if } x > 0 \\ -x \text{ if } x \leq 0 \end{cases}$
  - Code: `if` / `else` structure.
- **Derivative** (Numerical Approximation):
  - Math: $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$
  - Code: `(f(x + h) - f(x)) / h` where h is very small.

Linear Algebra (Matrices & Vectors)

- **Vectors**: Typically implemented as standard lists `[1, 2, 3]` or NumPy arrays `np.array([1, 2, 3])`.
- **Matrices**: Nested lists `[[1, 2], [3, 4]]` or `np.matrix`/`np.array`.
- **Dot Product**:
  - Math: $\boldsymbol{a} \cdot \boldsymbol{b} = \sum a_i b_i$
  - Code: `np.dot(a, b)` or `a @ b` (Python 3.5+).
- **Transpose**:
  - Math: $\boldsymbol{A}^T$
  - Code: `A.T` or `np.transpose(A)`.

Probability and Statistics

- **Mean** (μ): `sum(data) / len(data)` or `np.mean(data)`.
- **Variance** (σ^2): `np.var(data)`.
- **Standard Deviation** (σ): `np.std(data)`.
- **Random Variable Selection**: `random.choice(data)` or `random.random()`.

Pro Tips

Precision Issues

Computers use floating-point arithmetic. Be careful with equality checks $0.1 + 0.2 \neq 0.3$ in code! Use `math.isclose()` for comparisons.

Vectorization

In languages like Python (with NumPy) or R, always prefer vectorized operations (e.g., `a + b` for two large arrays) over manual `for` loops for massive speed improvements.

Zero-based Indexing

Most programming languages start indexing at 0, while many math notations (and R/Julia) start at 1. Always double-check your loop boundaries!