Authored by Dhruva Raman

Cheatsheet

Preamble

This cheatsheet contains tabular references for terminology and notation. For detailed explanations, consult the internet. If these aren't enough, consult me!

Make a stab at memorising these immediately, but they won't stick and that's OK. Spaced learning is a good technique to facilitate this.

If I use a symbol / notation in lectures you've forgotten, *put your hand up and ask*. That helps remind everybody. Eventually they will come naturally.

Not all of these definitions are immediately important. Some (e.g functionals and operators) come later in the course.

Recommended approach:.
First read this really good/important guide on writing maths. As you encounter symbols you're unfamiliar with, use the reference tables below

Standard mathematical objects	
Variables	Symbols that *represent* objects. $x = 1$ *assigns* the symbol x to the object 1 (a number). x is now a variable representing 1 . The *value* of x is 1.
Expressions (e.g. $x^2 + 4$)	A combination of variables and other objects, combined through mathematical operations (addition, subtraction, etc). Expressions are often assigned to variable names.
Equations (e.g. $x^2 = 4 + y$)	Multiple expressions *constrained* by an equals sign. *Solving* an equation means finding what values of variables make an equation true
Arrays (e.g. $A = [1, 1,$ 😀 $, 2]$)	Ordered collections of elements. Duplicates allowed . Indices referred to with subscripts. So $A_3 =$ 😀
Sets, (e.g. $S = \{1, 2, \text{hi}\}$)	Unordered collections of elements. Usually encapsulated with curly brackets. So S_2 doesn't make sense as there is no second element in S
Functions	Objects that take inputs and provide outputs. The *domain* is the set of allowable inputs. The *range* is the set of allowable outputs.
Operator	Special type of function, where the domain and range are sets of functions. So an operator takes in a function, and spits out a function. We'll encounter the *differential operator* later.
Functional e.g. $O(f) = f(2)$	Special type of function that takes in a function, and spits out a number. The example spits out the input function evaluated at 2.

whether computational objects in Python, or mathematical objects in your mind

Good programmers often use functionals and operators (they are often called **higher-order** functions in programming). Aim to become a good programmer. Examples are interspersed in notebooks

Standard maths notation	
$\mathbb{R}, \mathbb{N}, \mathbb{Z}$	Special sets: the real numbers, natural numbers, and integers.
$[4, 6]$	The *closed* interval of numbers between 4 and 6 (closed means inclusive).
$(-3, 3)$	The *open* interval of numbers between −3 and 3 (open not inclusive).
$f : \mathbb{R} \to [-1, 1]$	f is a function that maps real numbers (domain) to the closed interval $[-1, 1]$ (range). EG $f = \sin$ (the sine)
$f \circ g : \mathbb{R} \to [-1, 1]$	$f \circ g$ *composes* the functions f and g. Apply g first to the input. Then f. *Not* the same as multiplication
sin, cos, tan,exp	Special functions you should know and graph for yourself. We will use exp a lot. The other less
2^3	2 to the power of 3 ($2 \times 2 \times 2$). 3 is the *exponent*. You should know the *laws of exponents*
$a \in A$	a is *in* A. So a is an element, and A is a collection of elements containing a
$a \notin A$	a is *not* in A.
$(-\infty, a)$	All the numbers between minus infinity and a, exclusive. So all the numbers less than a
|a|	The absolute value of a if a is a number. So $\lvert -7 \rvert = 7$
|A|	The *cardinality* of A if A is a collection of elements (e.g. a set). Cardinality means number of elements. So $\lvert A \rvert = 3$ if $A = \{1, 2, \text{orange}\}$
$a \Rightarrow b, a \Leftarrow b,$ $a \Leftrightarrow b$	a implies b, a is implied by b, a and b are equivalent (implied by each other)

Set notation and grammar	
$A \cap B$ (or $A \wedge B$)	Intersection of sets A and B. Elements that are in A **and** B
$A \cup B$ (or $A \vee B$)	Union of sets A and B. Elements that are in A **or** B
$\emptyset$	The *empty set*: a set containing no elements
A^c	The complement of A. Elements that are *not* in the set A
$A \setminus B$	A not B: elements in A but not in B
: (or |)…"*such that*"	So $\{a \in \mathbb{R} : a > 2\}$ means *the set of real numbers* a **such that** a *is greater than* 2, i.e. the set of all real numbers greater than 2
$\exists$ … "*there exists*"	$\exists a \in \mathbb{N} : a > 2$ is the statement : "*there exists a natural number* a *such that* a *is greater than 2*"
$\forall$ … "*for all*"	$x^2 > 0 \; \forall x \in \mathbb{R}$ means x *squared is greater than* 0 *for every* x *in the set of real numbers*.

Make sure you are happy with the following notebook on sets.
They are important.

Inferring notation from context

$A(x)$	One set of round brackets after A tells me that A is a function, which accepts as an input whatever is in the round brackets. $A(x)$ is the output of the function A given input x
$A(x, y)$	Same as above, but A has two inputs: x and y
$A(x)$, where $x \in \mathbb{R}^n$	A has n inputs, but they are all packed in a single variable x
$f(x)g(x)$	$f(x) \times g(x)$. f and g are functions due to round brackets (see above). But mathematicians often/usually omit explicit multiplication signs.
$(x^2 + 4)(y^2 + 3)$	Despite the round brackets, there are no functions denoted. Two sets of round brackets means you multiply what's in each set of brackets. One set means what's in the brackets are inputs to a function, as in the previous examples.
$(1 + 3) \div 4$	Remember BODMAS. Brackets first
`A[x]`	In programming languages, square brackets usually mean *indices*. So `A` is an ordered collection, and `x` is the index
`A(x)`	In programming languages, like in maths, round brackets denote functions.

A_4 A_i	The subscript means A is an *ordered* collection of elements. A_4 is the fourth element. A_i is the i^{th} element, where $i \in \mathbb{N}$. EG $i = 3$.
$A(4)$ $A(x)$ $A(4, 5)$	The round brackets mean A is a function $A(4)$ is the output of the function with input 4 $A(x)$ is with input x $A(4, 5)$ means the function takes two inputs
$\sum_{i=1}^{N} A_i$, $\sum_{\omega \in \Omega} A(\omega)$	$\sum$ means sum. So add elements $A_1 + A_2 ... A_N$ $\in$ means Ω is a set. So add $A(\omega)$ for all elements ω in the set.

- The grammar section above allows you to build sets and arrays based on conditions. So $S = \{x \in \mathbb{R} : x > 4\}$ should be understandable and readable to you. This is known as *set builder* notation.

Test yourself

Go through the examples on wikipedia entry and make sure you can read them.

Practice the quiz here.

Pragmatic writing

Practice writing mathematical notation (like the symbols above) in markdown cells in your notebook of choice (Marimo/Pluto/Jupyter). These all use "LaTeX notation"….

LaTeX itself is a complete, professional document writing system. Learn to use LaTeX to write all your documents. Ideally, install LaTeX on your computer. But it takes up loads of space. Instructions here.

If you don't want to install LaTeX, open an account on overleaf.com to write in the cloud. Overleaf also has good tutorials.

This document was made with Typst, which is a simpler, newer, better (in my opinion) version of LaTeX. But it's nonstandard and has fewer tutorials/templates/etc so I don't recommend it. I'm just a hipster.

Why not use Word or google docs? Because LaTeX documents are standard for writing and sharing between people who write maths. Once you learn it, it's quicker, easier, and more flexible. Especially once you hit bibliography management on your dissertation…