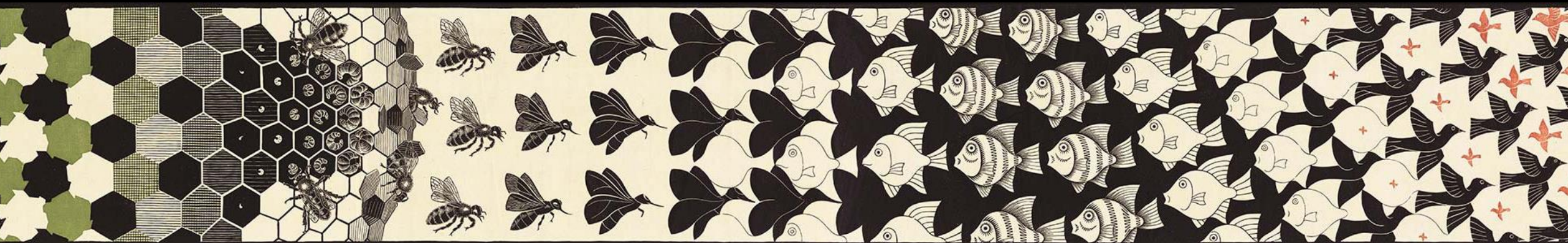


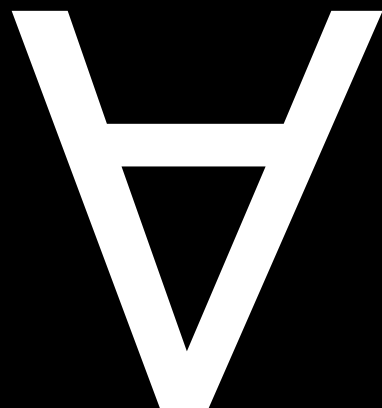
Data, Math and Methods

Week 3, Operators & Logic

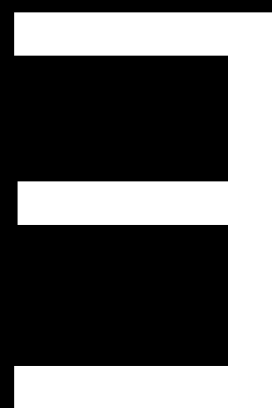


Today

- Mathematic notation
- Operation properties
- Paper: Logic
- Code: Drawing a circle



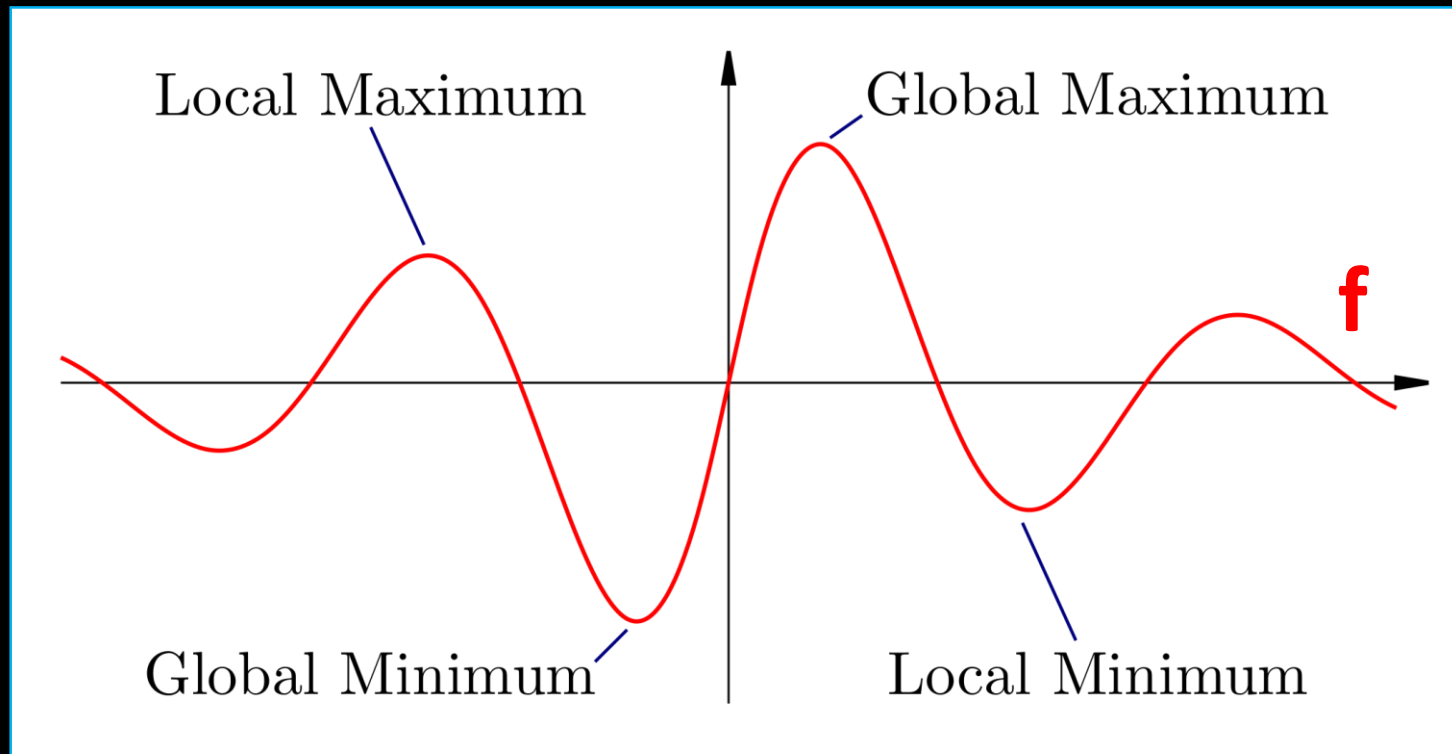
For all



Exists

Mathematical notation

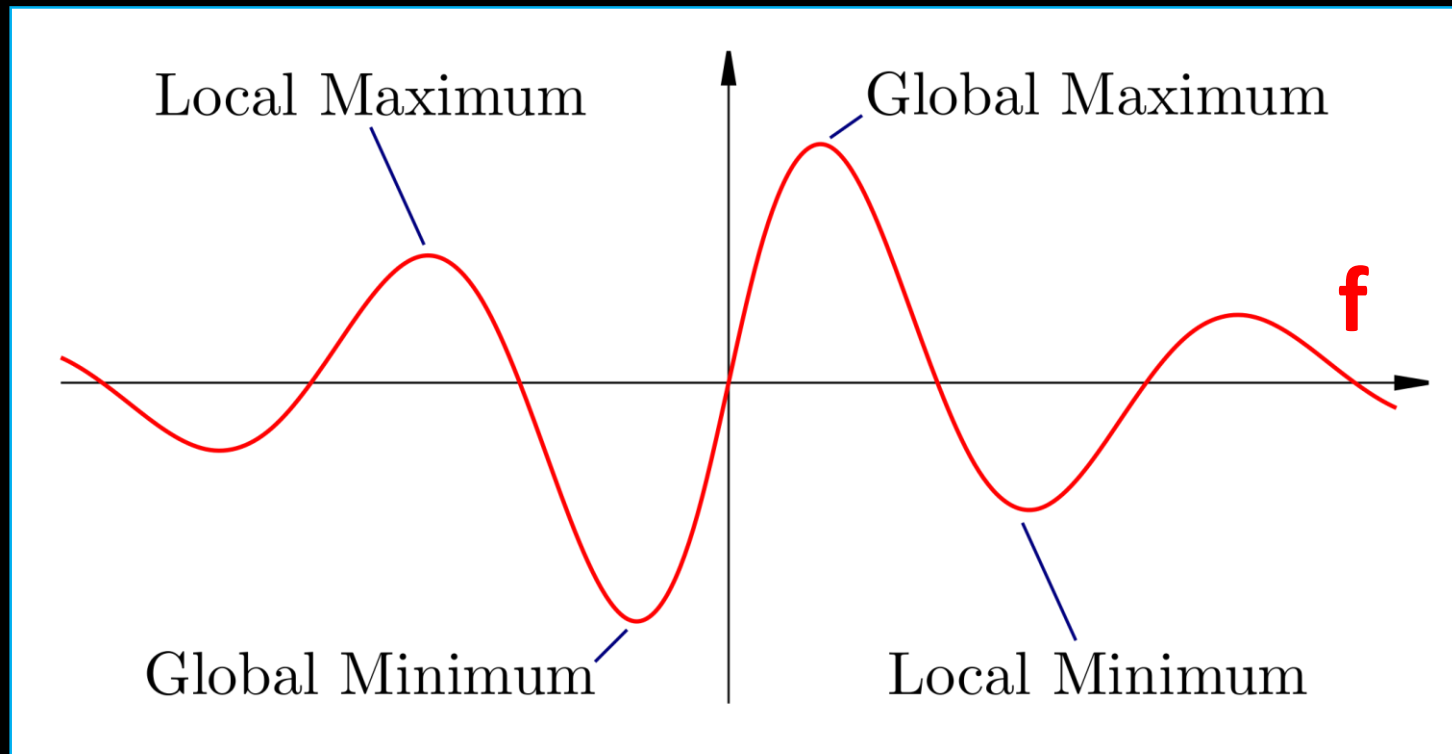
- Let's define something relatively simple using that notation:



Local maximum
Global maximum

Mathematical notation

- Let's define something relatively simple using that notation:



Local maximum
Global maximum

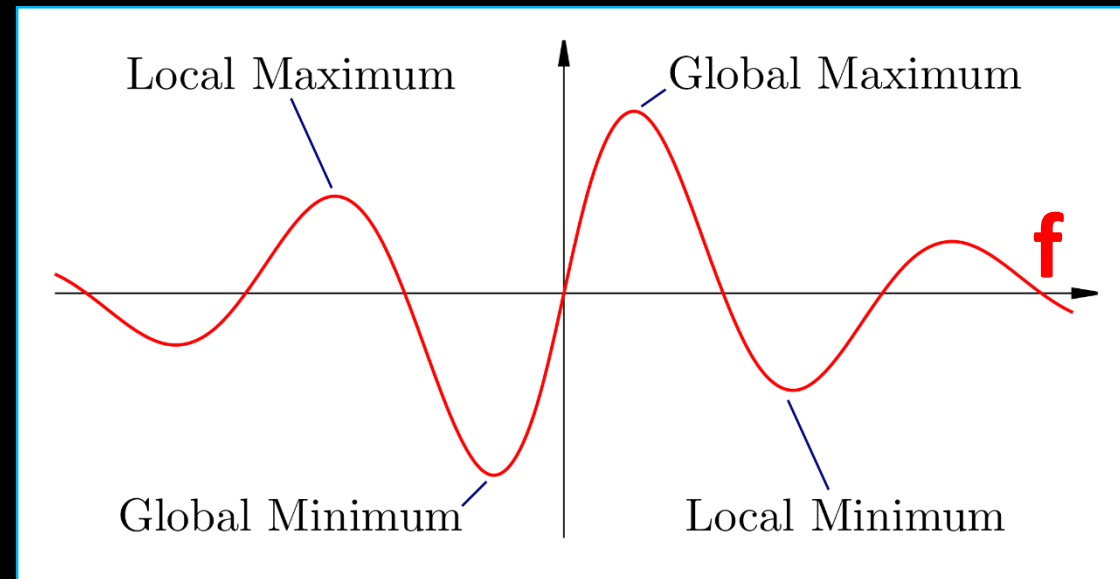
Let's try
explaining /
defining one of
them ...

Function $f: X \rightarrow \mathbb{R}$

\mathbb{R} ... Domain of all Real numbers

X ... Domain of all numbers we can put into the function (imagine it also as real numbers).

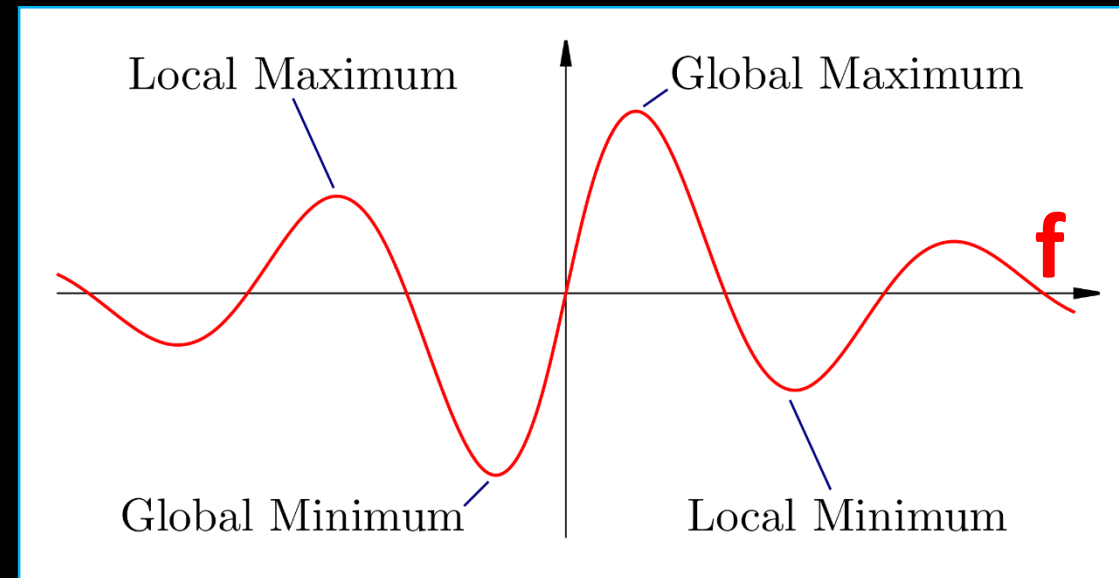
Number $c \in X$... a number from that domain.



Function $f: X \rightarrow \mathbb{R}$

\mathbb{R} ... Domain of all Real numbers

X ... Domain of all numbers we can put into the function (imagine it also as real numbers).



Number $c \in X$... a number from that domain.

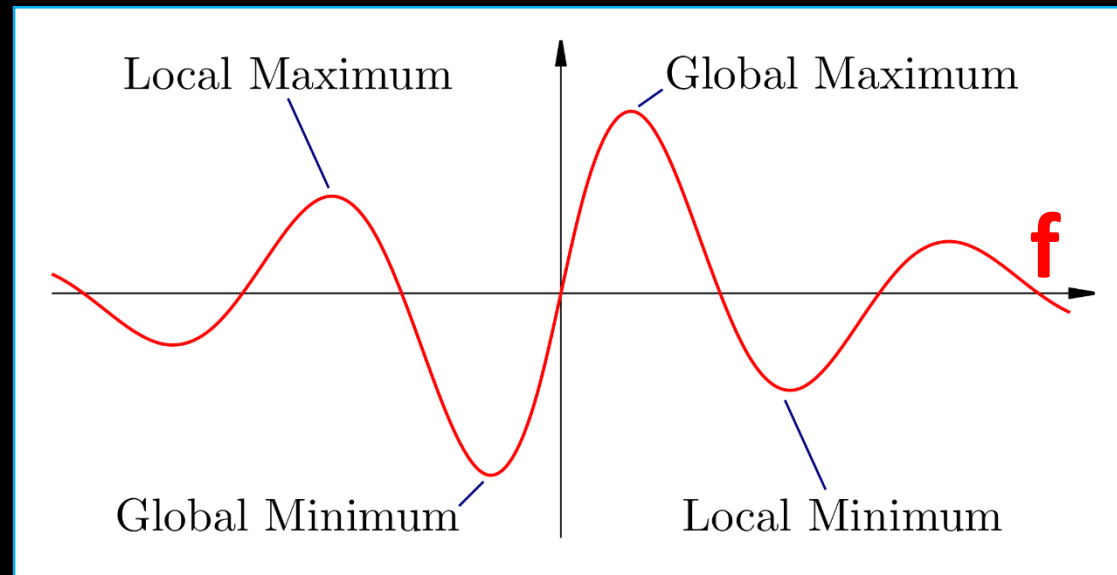
Definition:

Let $c \in X$ be the global maximum point of a function $f: X \rightarrow \mathbb{R}$
if:

Function $f: X \rightarrow \mathbb{R}$

\mathbb{R} ... Domain of all Real numbers

X ... Domain of all numbers we can put into the function (imagine it also as real numbers).



Number $c \in X$... a number from that domain.

Definition:

Let $c \in X$ be the global maximum point of a function $f: X \rightarrow \mathbb{R}$

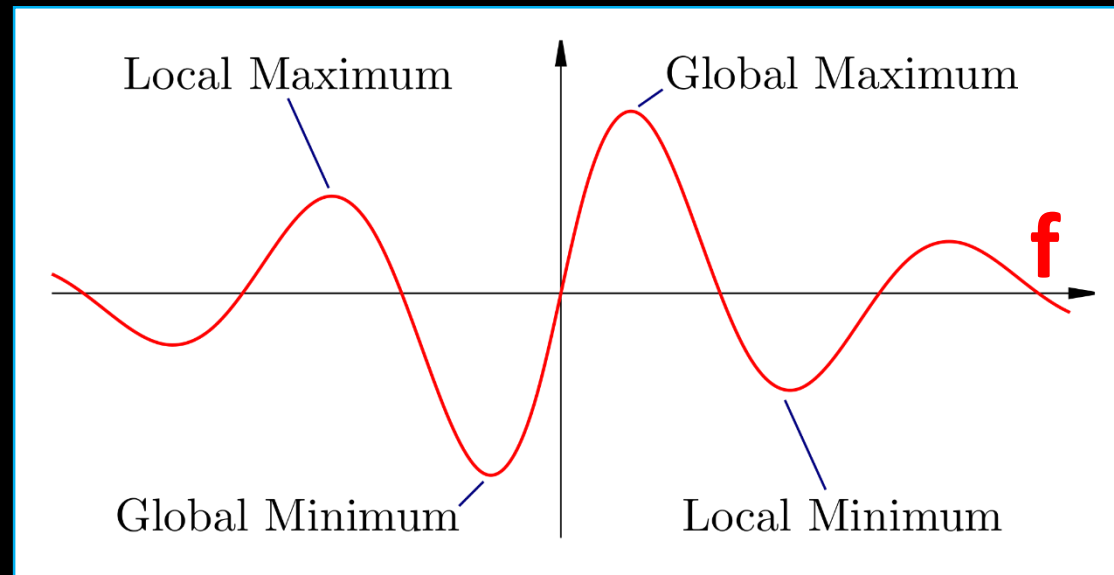
if: $\forall x \in X$

for all numbers

Function $f: X \rightarrow \mathbb{R}$

\mathbb{R} ... Domain of all Real numbers

X ... Domain of all numbers we can put into the function (imagine it also as real numbers).



Number $c \in X$... a number from that domain.

Definition:

Let $c \in X$ be the global maximum point of a function $f: X \rightarrow \mathbb{R}$

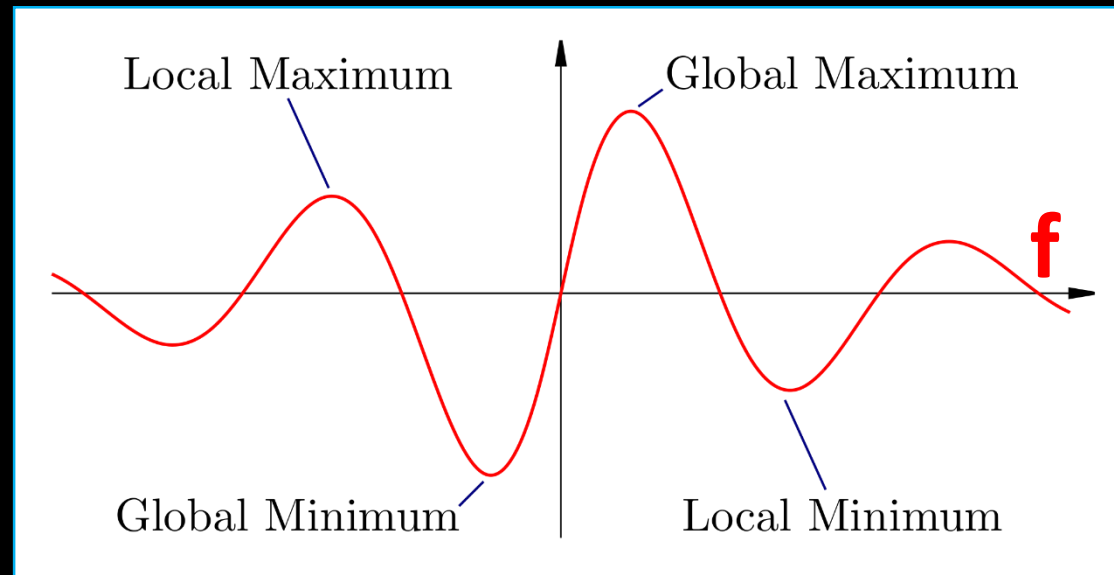
if: $\forall x \in X$ this is true: $f(c) \geq f(x)$.

for all numbers

Function $f: X \rightarrow \mathbb{R}$

\mathbb{R} ... Domain of all Real numbers

X ... Domain of all numbers we can put into the function (imagine it also as real numbers).



Number $c \in X$... a number from that domain.

Definition:

Let $c \in X$ be the global maximum point of a function $f: X \rightarrow \mathbb{R}$

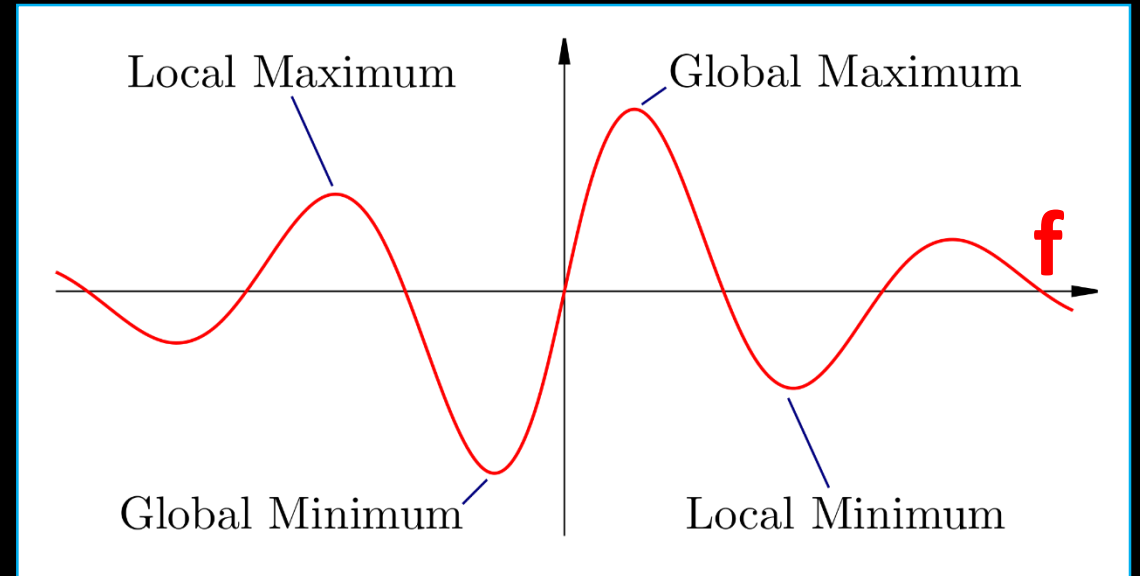
if: $\forall x \in X$ this is true:

for all numbers

$$f(c) \geq f(x) .$$

this point gives larger function value than any of the other numbers we try.

Global maximum



Definition:

Let $c \in X$ be the global maximum point of a function $f: X \rightarrow \mathbb{R}$

if: $\forall x \in X: f(c) \geq f(x)$.



In mathematics we lay brick by brick the basic building blocks (definitions) to then talk about more complicated concepts.

Mathematical notation helps us write in concise way in shared abstract language.

Operators

- What is an operator?

+

*

-

%

not ()

sin ()

cos ()

...

Operators

- What is an operator?

+

*

-

%

not ()

sin ()

cos ()

...

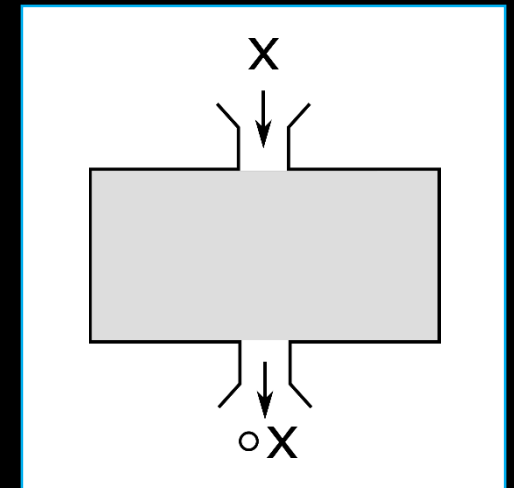
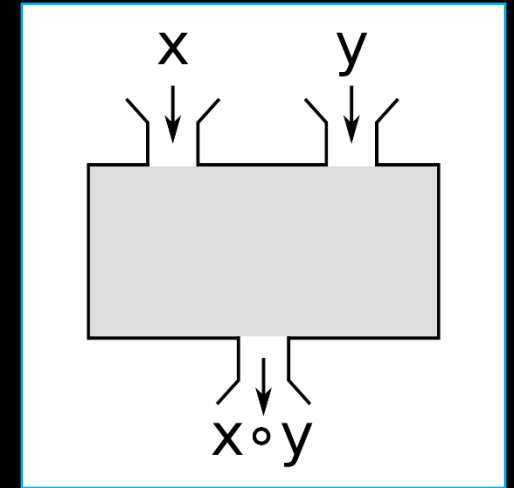
But rather than calculating some basic examples using these (which would be easy), let's talk about the rules and properties they have...

Operators

- There are rules about priority of evaluation:
For example:
 - $*$, $\%$ would be first
 - $+$, $-$ would go after
- Sometimes order matters, sometimes not; this also influences how we can reformat the whole formula (simplify).

Operators

- Binary operation
 - Takes two inputs
 - For example: $a+b$, $a*b$
- Unary operation
 - Takes a single input
 - For example: $\text{not}(a)$, $\sin(a)$



Operators properties

- Associative property:

A binary operation \blacksquare is called associative if:

$$\forall a, b, c \in \mathbb{R}: (a \blacksquare b) \blacksquare c = a \blacksquare (b \blacksquare c)$$

Operators properties

- **Associative property:**

A binary operation $*$ is called associative if:

$$\forall a, b, c \in \mathbb{R}: (a * b) * c = a * (b * c)$$

If: for any three numbers we can just *willy nilly* swap the brackets like this

For example:

- This is true for:

Operators properties

- **Associative property:**

A binary operation $*$ is called associative if:

$$\forall a, b, c \in \mathbb{R}: (a * b) * c = a * (b * c)$$

If: for any three numbers we can just *willy nilly* swap the brackets like this

For example:

- This is true for: $*$ multiplication, $+$ addition
- This is not true for:

Operators properties

- **Associative property:**

A binary operation $*$ is called associative if:

$$\forall a, b, c \in \mathbb{R}: (a * b) * c = a * (b * c)$$

If: for any three numbers we can just *willy nilly* swap the brackets like this

For example:

- This is true for: $*$ multiplication, $+$ addition
- This is not true for: $-$ subtraction

Operators properties

- **Associative property:**

A binary operation $*$ is called associative if:

$$\forall a, b, c \in \mathbb{R}: (a * b) * c = a * (b * c)$$

As a function:

Function f is associative if: $f(f(x, y), z) = f(x, f(y, z))$

Operators properties

- Commutative property:

A binary operation \blacksquare is called commutative if:

$$\forall a, b \in \mathbb{R}: a \blacksquare b = b \blacksquare a$$

Operators properties

- Commutative property:

A binary operation $*$ is called commutative if:

$$\forall a, b \in \mathbb{R}: a * b = b * a$$

Operators properties

- **Commutative property:**

A binary operation $*$ is called commutative if:

$$\forall a, b \in \mathbb{R}: a * b = b * a$$

As a function:

Function f is commutative if: $f(a, b) = f(b, a)$

Operators properties

- Operation **Addition** (+) is *associative* and *commutative*.
 - Associative $\forall a, b, c \in \mathbb{R}: (a + b) + c = a + (b + c)$
 - Commutative $\forall a, b \in \mathbb{R}: a + b = b + a$

Operators properties

- Why do we care about these properties?
 - Given some them, we can say something about the operator in general. For example if we can swap it around in the formula / if we can collapse it into brackets and overall simplify the problem.

Lesson?

- Lesson of this part is ...
 - ... that Math includes definitions of even the smallest building blocks (*who needs to describe behavior of + ?*)
 - ... *kinda* allows modularity/re-definition of these basic steps
 - ... builds from bottom up (definitions of the smallest particle → slowly describing more complex composites)

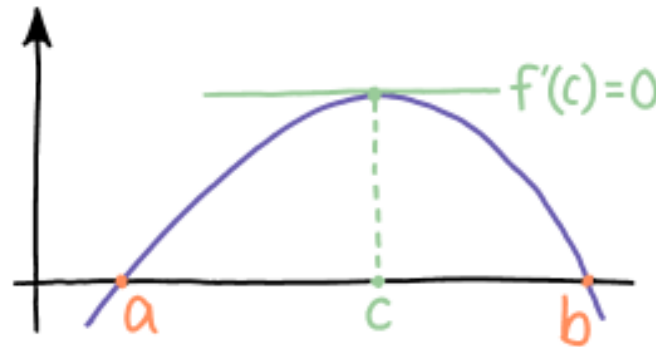


Pause 1

ROLLE'S THEOREM

FROM WIKIPEDIA, THE FREE ENCYCLOPEDIA

ROLLE'S THEOREM STATES THAT ANY REAL, DIFFERENTIABLE FUNCTION THAT HAS THE SAME VALUE AT TWO DIFFERENT POINTS MUST HAVE AT LEAST ONE "STATIONARY POINT" BETWEEN THEM WHERE THE SLOPE IS ZERO.



EVERY NOW AND THEN, I FEEL LIKE THE MATH EQUIVALENT OF THE CLUELESS ART MUSEUM VISITOR SQUINTING AT A PAINTING AND SAYING "C'MON, MY KID COULD MAKE THAT."

Logic

- And, Or, Not, Xor, \Rightarrow , \Leftrightarrow ...
- Logic tables for atoms
- Logic tables for whole formulas

Logic formula

- Conversion of formulas
- Building specific logic operators from combination of other operators (aka what if we only have AND and OR at our disposal, but we want to create NAND)

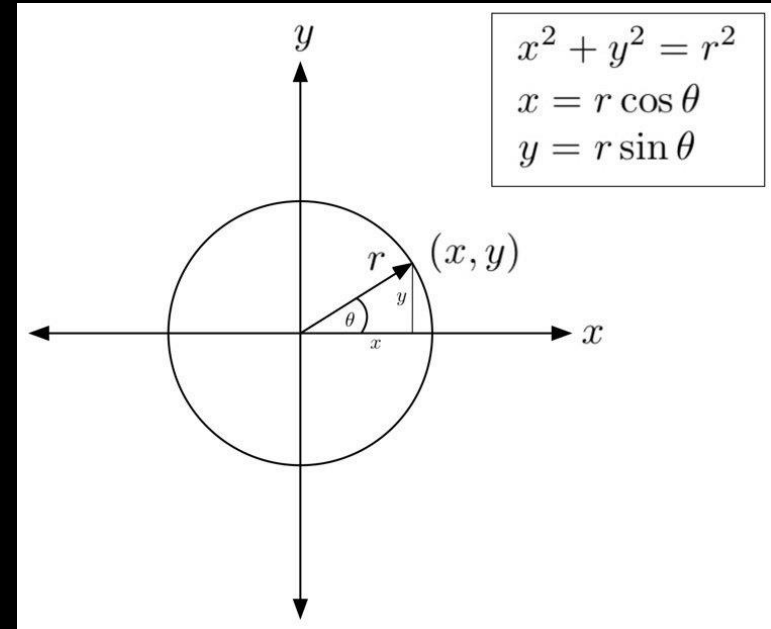
Pause 2

Code: circle

- Inspiration from the demo at: http://www.generative-gestaltung.de/2/sketches/?01_P/P_2_2_3_01

Task: points on circle

- Points on circle given by a formula:
 - Get (x,y) from the parameter of angle and radius



- Task – make a python code which would draw a circle (drawing it point by point, don't use functions to draw it directly)
- Starter code: (TBA)