Infix, Postfix, Prefix Notations

operand: The character or the number that the operation is going to done on.

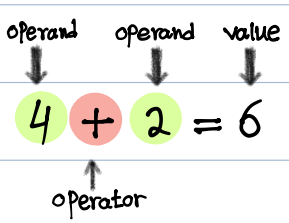
Ex: 1 'a' 100 'abc'

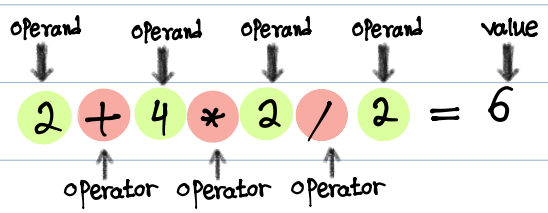
operator: A symbol that is going to represent an operation to be done on an operand.

Ex: + — * / ^

Expression: A sequence of operands and operations that when evaluated, produces a single value.

Ex:





Infix, Postfix, Prefix Notations are used to represent mathematical expressions, but the order of operands, operators are different in each of them.

Infix: operators are placed between the operands

operand1	operator	operand2
4	+	2

Postfix: operators are placed after the operands.

operand1	operand2	operator
4	2	+

Prefix: operators are placed before the operands.

operator	operand1	operand2
+	4	2

All of the above notations means the same thing, $4+2=6$

Advantages: Prefix and postfix don't need to follow precedence or associative rule.

operator	Associative Rule	Precedence
^	Right to left	Highest
*, /	Left to right	Higher
+, −	Left to right	Lowest

Precedence: It shows which operation should be done first; an operator with highest precedence should be performed first.

Associative: It specifies from which direction the operations should be performed.

Examples

Precedence: $4 + 2/2 = 5$

Assciative (L→R): $8 + 2 - 9 = 1$

Assciative (R→L): $2 \wedge 1 \wedge 10$

$1 \wedge 10 = 1$

$2 \wedge 1 = 1$

Infix to Postfix Conversion: Parenthesis all the operands around the operators.

start converting to postfix from operators with highest Priority

Ex: $X + y * z \Rightarrow (x + (y * z)) \Rightarrow (x + (yz*)) \Rightarrow Xyz*+$

Ex: $(X+y) * z \Rightarrow ((X+y) * z) \Rightarrow ((Xy+) * z) \Rightarrow Xy+z*$

Infix to Prefix Conversion:

- parenthesise every expression
- start conversion from the operators with highest Precedence

Ex: $X + y * z \Rightarrow (x + (y * z)) \Rightarrow (X + (*yz)) \Rightarrow +X*yz$

Ex: $(X+y) * z \Rightarrow ((x+y) * z) \Rightarrow (+xy) * z) \Rightarrow *+Xyz$