

Infix, Postfix, Prefix

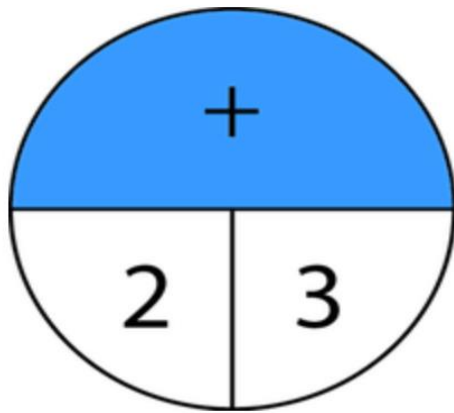
Introduction Part-2

Terminology

- Infix: The notation commonly used in mathematical formula
- Operand: The value on which an operator is performed
- Operator: A symbol like minus that shows an operation
- Postfix: A mathematical notation in which operators follow operands
- Prefix: A mathematical notation in which operands follow operators
- Reverse Polish Notation (RPN): A mathematical notation in which operators follow operands

Prefix

- Prefix notation မှာ operands တွေရဲ့ရှေ့မှာ operator ရှိပါတယ်။ ကျွန်တော်တို့ infix နဲ့ prefix နှိုင်းယှဉ်ကြည့်ပါမည်။
- Eg: $2 + 3$ ဆိုတဲ့ infix notation ကို ကျွန်တော်တို့ prefix notation ပုံစံပြောင်းလဲကြည့်ရအောင်
- **The Prefix Example of 2+3**

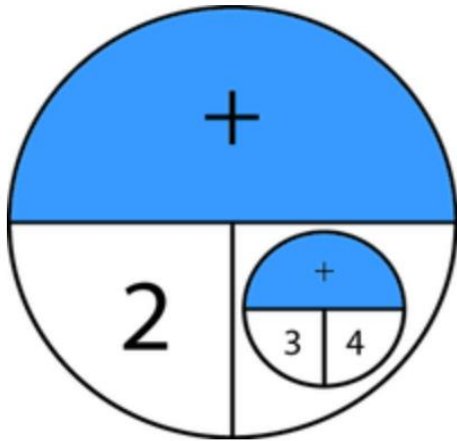


- စက်ဝိုင်းပုံ ထဲက blue section မှာ ဆို operator ကိုထည့်ထားတာပဲဖြစ်ပါတယ်။ Operator + ကို ယူပြီးတာနဲ့ operand နှစ်လုံးကို left-to-right order အတိုင်း read လုပ်သွားမှာဖြစ်ပါတယ်။ **Prefix: + 23** ဆိုပြီးရလာမှာပါ။

multiple operators (prefix)

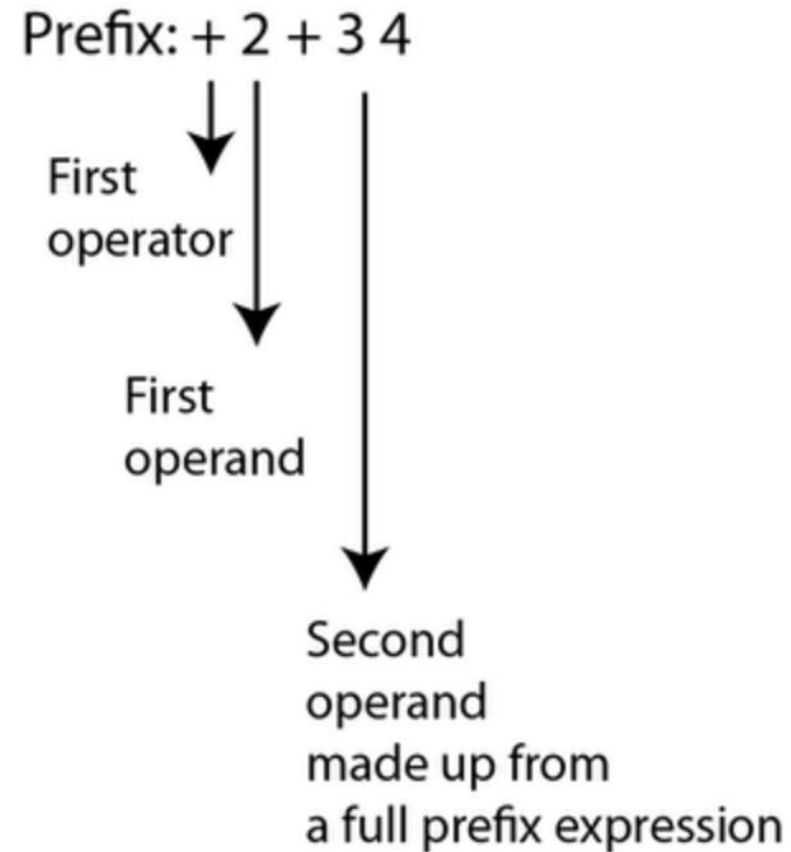
- ကျွန်တော်တို့ expression ထဲမှာ operators တစ်ခုထက်ပိုပါဝင်တဲ့ $2 + 3 + 4$ Infix ပုံစံကို prefix ပြောင်းကြည့်မှာဖြစ်ပါတယ်။

- **The Prefix Example of 2+3+4**



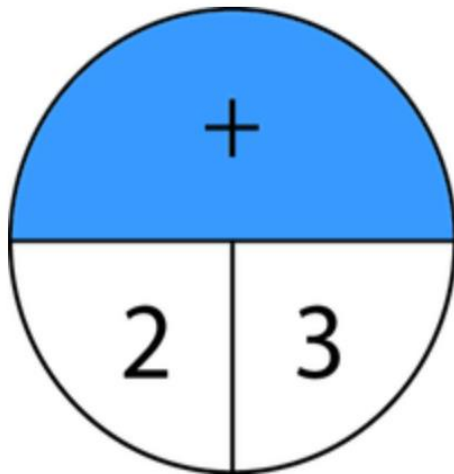
- ဒီပုံမှာဆို outer circle ထဲမှာ inner circle တစ်ခုပါဝင်လာပြီး Outer circle ကိုအရင် read လုပ်ပြီးမှ inner circle ကို read လုပ်မှာဖြစ်ပါတယ်။ + operator ကိုအရင်ယူမယ်ပြီးရင် operand တွေကို left-to-right order အတိုင်း read လုပ်သွားမှာဖြစ်လို့ left မှာရှိနေတဲ့ 2 ကိုအရင် read လုပ်မယ် ပြီးရင် inner circle ကို read လုပ်မှာဖြစ်ပါတယ်။ inner circle ကိုလည်းပုံမှန် rule အတိုင်း အလုပ်လုပ်သွားမှာဖြစ်ပါတယ်။ + operator ကို read မယ်ပြီးရင် 3 နဲ့ 4 ကို left to right order အတိုင်း read လုပ်သွားမှာဖြစ်ပါတယ်။ **Prefix : + 2 + 34** ဆိုပြီး prefix ပုံစံအတိုင်းရလာမှာဖြစ်ပါတယ်။

Prefix: + 2 + 3 4



Postfix OR Reverse Polish Notation

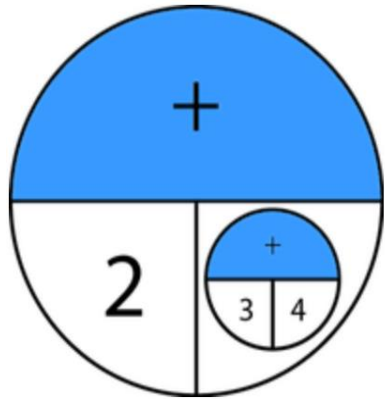
- Postfix notation ကို Reverse Polish notation လိုလည်းခေါ်ပါတယ်။ Postfix notation ကတော့ operator ကို operands တွေပြီးမှထည့်ပေးတာပဲဖြစ်ပါတယ်။
- Eg: $2 + 3$ ဆိုတဲ့ infix notation ကို ကျွန်တော်တို့ postfix notation ပုံစံပြောင်းလဲကြည့်ရအောင်
- **The Postfix Example of 2+3**



- Postfix မှာတော့ operator ကို operand တွေပြီးမှ put လုပ်မှာမို့လို့ 2 နဲ့ 3 ဆိုတဲ့ operand 2ခုကို left-to-right order အတိုင်း read လုပ်ပြီးမှ + operator ကို read လုပ်မှာဖြစ်ပါတယ်။ **Postfix : 23 + ဆိုပြီးရလာမှာဖြစ်ပါတယ်။**

multiple operators (postfix)

- ကျွန်တော်တို့ expression ထဲမှာ operators တစ်ခုထက်ပိုပါဝင်တဲ့ $2 + 3 + 4$ Infix ပုံစံကို postfix ပြောင်းကြည့်မှာဖြစ်ပါတယ်။
- ကျွန်တော်တို့သိထားရမှာက operator ကို operand နောက်မှာထည့်ရမယ်ဆိုတာပါ။



- Postfix မှာ operand တွေကို အရင် read လုပ်မှာ 2 ကိုအရင် read လုပ်မှာဖြစ်ပါတယ်။ပြီးရင် inner circle ထဲက operand 2 ခု ဖြစ်တဲ့ 3 နဲ့ 4 ကို left to right order အတိုင်း ယူပြီးမှ + operator 2 ခုကို operand တွေရဲ့နောက်မှာထည့်ပေးရမှာပဲဖြစ်ပါတယ်။ **Postfix : 2 3 4 ++** ဆိုပြီး postfix ပုံစံအတိုင်းရလာမှာဖြစ်ပါတယ်။
- **Infix : 2 + 3 + 4 → Postfix : 2 3 4 ++**

Why do we need 3 different expression formats?

- Infix expressions are human readable but not efficient for machine reading.
- Prefix and Postfix do not need the concept of precedence & associativity hence it become highly efficient to parse expressions in prefix or postfix formats.

အခုလိုမျိုး tips လေးတွေကို ကျွန်တော်တို့ facebook page ဖြစ်တဲ့ code craft မှနေ့စဉ်တင်ပေးမှာဖြစ်တာကြောင့် like and follow လေးတွေလုပ်ထားပေးကြပါခဗျာ။

Source from (Steven Curtis)

အားလုံးကို ကျေးဇူးတင်ပါတယ်ခဗျာ။ <Code craft>