# Deletion at beginning

Deletion in doubly linked list at the beginning is the simplest operation. We just need to copy the head pointer to pointer ptr and shift the head pointer to its next.

```
Ptr = head;
   head = head → next;
```

now make the prev of this new head node point to NULL. This will be done by using the following statements.

```
head → prev = NULL
```

Now free the pointer ptr by using the **free** function.

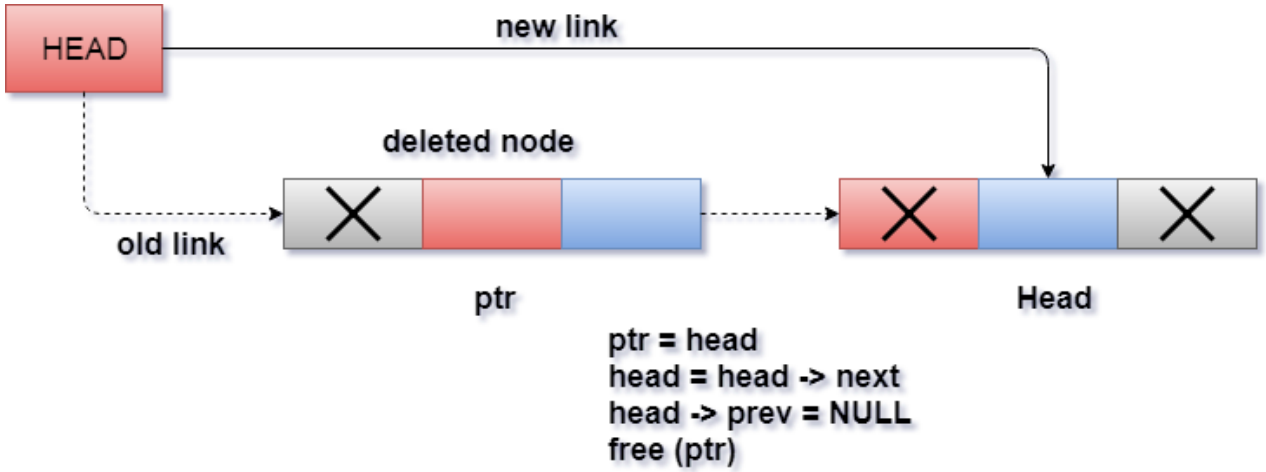
```
free(ptr)
```

# Algorithm

- **STEP 1:** IF HEAD = NULL

  WRITE UNDERFLOW
  GOTO STEP 6

- **STEP 2:** SET PTR = HEAD
- **STEP 3:** SET HEAD = HEAD → NEXT
- **STEP 4:** SET HEAD → PREV = NULL
- **STEP 5:** FREE PTR
- **STEP 6:** EXIT




**Deletion in doubly linked list from beginning**

## C Function

```
#include<stdio.h>
#include<stdlib.h>
void create(int);
void beginning_delete();
struct node
{
```

```
    int data;
    struct node *next;
    struct node *prev;
};
struct node *head;
void main ()
{
    int choice,item;
    do
    {
        printf("1.Append List\n2.Delete node from beginning\n3.Exit\n4.Enter your choice?");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            printf("\nEnter the item\n");
            scanf("%d",&item);
            create(item);
            break;
            case 2:
            beginning_delete();
            break;
            case 3:
            exit(0);
            break;
            default:
            printf("\nPlease enter valid choice\n");
        }
```

```
    }while(choice != 3);
}
void create(int item)
{

   struct node *ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
      printf("\nOVERFLOW\n");
   }
   else
   {



   if(head==NULL)
   {
      ptr->next = NULL;
      ptr->prev=NULL;
      ptr->data=item;
      head=ptr;
   }
   else
   {
      ptr->data=item;printf("\nPress 0 to insert more ?\n");
      ptr->prev=NULL;
      ptr->next = head;
      head->prev=ptr;
      head=ptr;
   }
```

```
    printf("\nNode Inserted\n");
}

}
void beginning_delete()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\n UNDERFLOW\n");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nNode Deleted\n");
    }
    else
    {
        ptr = head;
        head = head -> next;
        head -> prev = NULL;
        free(ptr);
        printf("\nNode Deleted\n");
    }
}
```

**Output**

```
1.Append List
2.Delete node from beginning
3.Exit
4.Enter your choice?1

Enter the item
12

Node Inserted
1.Append List
2.Delete node from beginning
3.Exit
4.Enter your choice?2

Node Deleted
1.Append List
2.Delete node from beginning
3.Exit
4.Enter your choice?
```

← prev next →

## Please Share

## Learn Latest Tutorials


Swift

Pig

Flask

C. Graphics


Automata

Testing

## Preparation

Aptitude

Reasoning


Verbal A.

Interview

## B.Tech / MCA


DBMS


DS


DAA

OS

C. Network

Compiler D.

COA


D. Math.

E. Hacking


Web Tech.

Cyber Sec.

C

C++

Java


.Net

Python


Programs


Control S.

1. Artificial Intelligence Programs
2. Python Tutorial PDF
3. Artificial Intelligence Courses
4. How To Create Your Own Website