

تفاوت‌های اصلی بین heap و stack در مدیریت حافظه، نحوه تخصیص و آزادسازی حافظه، و کاربردهای معمول آن‌ها در برنامه‌نویسی به شرح زیر است:

## ( Stack پشته )

### 1. مدیریت حافظه :

- Stack به صورت LIFO (Last In, First Out) مدیریت می‌شود، یعنی آخرین داده‌ای که وارد شده است، اولین داده‌ای است که خارج می‌شود.

- تخصیص و آزادسازی حافظه در Stack به صورت خودکار و توسط کامپایلر انجام می‌شود.

### 2. سرعت :

- عملیات در Stack بسیار سریع است زیرا مدیریت حافظه به صورت خودکار و ساده انجام می‌شود.

### 3. اندازه :

- اندازه Stack محدود و معمولاً توسط سیستم یا محیط برنامه‌نویسی تعیین می‌شود. بنابراین، برای داده‌های بزرگ یا تعداد زیادی داده ممکن است مناسب نباشد.

### 4. کاربردها :

- برای ذخیره‌سازی متغیرهای محلی، پارامترهای ورودی توابع و نگهداری آدرس بازگشتی توابع استفاده می‌شود.

### 5. طول عمر داده‌ها :

- داده‌ها در Stack معمولاً طول عمر کوتاهی دارند و با خروج از بلوک کد (مانند پایان یافتن یک تابع) آزاد می‌شوند.

## ( Heap توده )

### 1. مدیریت حافظه :

- Heap برای تخصیص حافظه به صورت پویا (دینامیک) استفاده می‌شود. برنامه‌نویس می‌تواند در هر زمان حافظه را تخصیص دهد و باید آن را به صورت دستی آزاد کند.

### 2. سرعت :

- عملیات در Heap کندتر از Stack است زیرا مدیریت حافظه پیچیده‌تر و تخصیص/آزادسازی آن زمان بیشتری می‌برد.

### 3. اندازه :

- اندازه Heap بزرگتر از Stack است و محدودیت اندازه کمتری دارد. بنابراین، برای داده‌های بزرگ یا زمانی که حجم زیادی از داده‌ها نیاز به ذخیره‌سازی دارند مناسب است.

### 4. کاربردها :

- برای ذخیره‌سازی داده‌هایی که طول عمر نامشخص یا طولانی دارند، مانند اشیاء (objects) و داده‌هایی که نیاز به تخصیص پویا دارند استفاده می‌شود.

### 5. طول عمر داده‌ها :

- داده‌ها در Heap می‌توانند طول عمر بیشتری داشته باشند و حتی پس از خروج از یک بلوک کد باقی بمانند تا زمانی که به صورت دستی آزاد شوند یا برنامه به پایان برسد.

جمع‌بندی تفاوت‌ها

- مدیریت حافظه : Stack خودکار و توسط کامپایلر، Heap دستی و توسط برنامه‌نویس.
- سرعت : Stack سریع‌تر، Heap کندتر.
- اندازه : Stack محدود، Heap بزرگتر.
- کاربردها : Stack برای داده‌های کوتاه‌مدت و محلی، Heap برای داده‌های بلندمدت و بزرگ.
- طول عمر داده‌ها : Stack کوتاه‌مدت، Heap بلندمدت.