**Difference Between Primitive Types and Wrapper Classes in Java**

In Java, every primitive data type has a corresponding **wrapper class** in the java.lang package. Wrapper classes are objects that encapsulate primitive values.

**1. What are Primitive Data Types?**

* **Primitive types** are the basic built-in data types in Java.
* They store simple values **directly in memory (stack)**.
* They **do not have methods** and cannot be null.

**Example of Primitive Types:**

int num = 10;

double price = 99.99;

char letter = 'A';

boolean isJavaFun = true;

**2. What are Wrapper Classes?**

* **Wrapper classes** are object representations of primitive types.
* Stored in **heap memory**, which adds some overhead.
* Can be **null** and provide useful **utility methods**.

**Example of Wrapper Classes:**

Integer numObj = Integer.valueOf(10);

Double priceObj = Double.valueOf(99.99);

Character letterObj = Character.valueOf('A');

Boolean isJavaFunObj = Boolean.valueOf(true);

**3. Key Differences**

| **Feature** | **Primitive Type** | **Wrapper Class** |
| --- | --- | --- |
| **Type** | Basic data type | Object (Class) |
| **Storage** | Stored in **stack memory** (faster) | Stored in **heap memory** (slower) |
| **Default Value** | Depends on type (int=0, boolean=false) | null (if not assigned) |
| **Can be null?** | ❌ No | ✅ Yes |
| **Methods Available?** | ❌ No | ✅ Yes (e.g., Integer.parseInt(), Double.valueOf()) |
| **Use in Collections?** | ❌ No (e.g., ArrayList<int> is invalid) | ✅ Yes (e.g., ArrayList<Integer>) |
| **Performance** | Faster (no object overhead) | Slower (extra object handling) |
| **Autoboxing Support?** | ❌ No | ✅ Yes |

**4. Autoboxing & Unboxing**

Java automatically converts between primitives and wrapper classes:

**Autoboxing (Primitive → Wrapper)**

int num = 10;

Integer numObj = num; // Autoboxing (int → Integer)

**Unboxing (Wrapper → Primitive)**

Integer numObj = Integer.valueOf(20);

int num = numObj; // Unboxing (Integer → int)

**5. When to Use What?**

| **Use Case** | **Recommended Type** |
| --- | --- |
| Performance-critical applications | **Primitive types** |
| Need to store in collections (ArrayList, HashMap) | **Wrapper classes** |
| Need null values | **Wrapper classes** |
| Need built-in methods (e.g., parsing, conversion) | **Wrapper classes** |

**Conclusion**

* **Use primitives** when you need speed and efficiency.
* **Use wrappers** when working with collections, null values, or need object methods.

Let me know if you need further clarification! 🚀