**TypeScript**

1. **Q:** What is the difference between interface and type in TypeScript?  
   **A:** interface can be extended, while type can define unions and intersections.
2. **Q:** How would you implement a generic function in TypeScript?  
   **A:** Use <T> to define a generic type and use it in the function signature.
3. **Q:** What is the difference between any and unknown?  
   **A:** any disables type checking, while unknown requires type assertion before use.
4. **Q:** How would you implement a decorator in TypeScript?  
   **A:** Create a function that takes a target and modifies it.
5. **Q:** What is the difference between readonly and const?  
   **A:** readonly is for properties, while const is for variables.
6. **Q:** How would you implement a mapped type in TypeScript?  
   **A:** Use keyof and in to create a new type based on an existing one.
7. **Q:** What is the difference between namespace and module?  
   **A:** namespace is for internal modules, while module is for external modules.
8. **Q:** How would you implement a conditional type in TypeScript?  
   **A:** Use extends and ? to define a type based on a condition.
9. **Q:** What is the difference between never and void?  
   **A:** never represents a function that never returns, while void represents a function that returns nothing.
10. **Q:** How would you implement a type guard in TypeScript?  
    **A:** Create a function that checks the type and returns a type predicate.
11. **Q:** What is the difference between abstract and interface?  
    **A:** abstract can have implementations, while interface cannot.
12. **Q:** How would you implement a utility type in TypeScript?  
    **A:** Use built-in utility types like Partial, Pick, or Record.
13. **Q:** What is the difference between as and ! in TypeScript?  
    **A:** as is for type assertion, while ! is for non-null assertion.
14. **Q:** How would you implement a mixin in TypeScript?  
    **A:** Use a function that extends a class with additional behavior.
15. **Q:** What is the difference between enum and const enum?  
    **A:** enum generates runtime code, while const enum is inlined at compile time.
16. **Q:** How would you implement a discriminated union in TypeScript?  
    **A:** Use a common property to distinguish between types in a union.
17. **Q:** What is the difference between keyof and typeof?  
    **A:** keyof gets the keys of a type, while typeof gets the type of a value.
18. **Q:** How would you implement a recursive type in TypeScript?  
    **A:** Define a type that references itself, e.g., type Tree<T> = { value: T; children: Tree<T>[] }.
19. **Q:** What is the difference between infer and extends?  
    **A:** infer extracts a type within a conditional type, while extends checks if a type satisfies a condition.
20. **Q:** How would you implement a type-safe dictionary in TypeScript?  
    **A:** Use Record<string, T> or { [key: string]: T }.