

RESPOSTAS

1. Qual a diferença entre tipagem dinâmica e tipagem estática?

Tipagem estática: a linguagem não permite que o programador altere o tipo da variável após ter sido declarada, tipagem dinâmica: capacidade e habilidade que a linguagem tem de alterar os tipos de dados de acordo com os valores atribuídos às variáveis.

2. Qual o principal problema do uso de tipagem dinâmica?

Podemos dizer que o principal problema do uso da tipagem dinâmica é que a detecção de erros só ocorre durante a execução, podendo gerar bugs e erros.

3. Pesquise um exemplo na internet em que a tipagem dinâmica pode ser problemático.

```
let valor = 42;  
console.log(valor + 10); // Saída: 52
```

```
valor = "Olá, mundo!";  
console.log(valor + 10); // Saída: "Olá, mundo!10"
```

Nesse exemplo, a tipagem dinâmica permite que a mesma variável "valor" seja usada para diferentes tipos de dados. Isso pode levar a resultados inesperados, como a concatenação de uma string e um número.

4. Pesquise e exemplifique com um exemplo porque dizemos que a linguagem C, mesmo tendo tipagem estática, possui tipagem fraca.

A linguagem tem tipagem estática pois a verificação é feita ainda em tempo de compilação, porém tem tipagem fraca por que permite conversão implícita entre valores de tipos diferentes, como no exemplo abaixo:

```
#include <stdio.h>
```

```
int main() {  
    int numeroInteiro = 42;  
    float numeroDecimal = 3.14;  
  
    // Atribuição de um float a uma variável int  
    numeroInteiro = numeroDecimal;  
  
    printf("Valor da variável inteira: %d\n", numeroInteiro); // Saída: 3  
  
    return 0;  
}
```

no exemplo vemos que a variável numeroInteiro foi definida como inteiro(sem vírgula) enquanto a outra variável foi definida como float(com casas decimais), no entanto ao igualarmos as variáveis, ela leu o valor de número inteiro normalmente, fazendo com que a mesma perdesse sua parte decimal

5. Pesquise e, se encontrar, um exemplo onde o tipo any seria benéfico.

-any é uma palavra chave usada para tipos dinâmico ou desconhecido, apesar de ser recomendado evitar o uso dela, existem casos que ela pode ser benéfica, como no exemplo abaixo:

```
// Dados da API externa (formato desconhecido)
```

```
const apiData: any = {  
  name: 'John Doe',  
  age: 25,  
  location: {  
    city: 'New York',  
    country: 'USA'  
  }  
};
```

```
// Acessando propriedades desconhecidas usando o tipo any
```

```
const personName: string = apiData.name;  
const personAge: number = apiData.age;  
const personCity: string = apiData.location.city;
```

```
console.log(personName, personAge, personCity);
```

Neste caso, como a estrutura dos dados da API é desconhecida, usar any permite que você acesse as propriedades sem erros de compilação.

6. Poderíamos dizer que a tipagem do TypeScript é fraca por uma variável do tipo number aceitar tanto inteiros como ponto flutuante?

Esse exemplo é na verdade de uma tipagem forte, o fato de ele aceitar tanto inteiro quanto flutuantes á uma variável do tipo number, significa que ele impõe restrições sobre os tipos de valores que podem ser atribuídos, fazendo com que somente valores numéricos possam ser atribuídos a variável.