The differences between arrow and traditional functions in JavaScript:

1. Syntax:

   Arrow functions have the "=>" syntax.

   Traditional functions have the "function" syntax.

   Example code:

   ```
   const arrowFunction = () => console.log(“Arrow function”);


   function traditinalFunction() {
           console.log(“Arrow function”);
   }
   ```

2. "This" keyword: Arrow functions do not have their own "this" keyword. They inherit "this" from the surrounding lexical scope. Traditional functions have their own "this" keyword which may change based on how the function is called. When called as a method, "this" refers to the object the method belongs to.

3. The "arguments" object: This is like an array containing all the arguments passed to the function. Arrow functions do not possess this object while traditional functions have it.

   Example code:

   ```
   function traditionalFunction() {
     console.log(arguments);
   }
   traditionalFunction(1, 2, 3);

   const arrowFunction = (...args) => {
     console.log(args);
   };
   arrowFunction(1, 2, 3);
   ```

4. "new" Keyword: Arrow functions cannot be used as constructors. Traditional functions can be used as constructors and can be called with the "new" keyword.

5. Prototype property: Arrow functions do not have a "prototype" property. Traditional functions have a "prototype" property, which is used for inheritance and the prototype chain in JavaScript.