

# WebAssembly

- امیرعلی سلیمی
- امیرنژاد ملایری
- علی قائم دوست

## WebAssembly چیست؟

WebAssembly یک فناوری نوین در حوزه وب است که به منظور اجرای کدهای بسیار سریع و بهینه در مرورگرهای اینترنتی ایجاد شده است. این فناوری برای اجرای کدهایی که در زبان‌هایی مانند C، C++، Rust و ... نوشته شده‌اند، استفاده می‌شود و این به برنامه‌های وب اجازه می‌دهد تا بهینه‌تر و سریع‌تر اجرا شوند. در این مقاله به بررسی اصول و کاربردهای WebAssembly، نحوه پیاده‌سازی آن و مثال‌هایی از کدهای WebAssembly می‌پردازیم.

## مفاهیم اولیه و کاربردهای WebAssembly

WebAssembly یک فرمت اجرایی است که برای اجرای کدهای کامپایل شده از زبان‌های برنامه‌نویسی مختلف استفاده می‌شود. این فناوری به برنامه‌های وب اجازه می‌دهد که به سرعت و با کارایی بالا اجرا شوند و به طور قابل توجهی بهبود کنند. برای مثال، با استفاده از WebAssembly می‌توان برنامه‌هایی را که قبلاً برای اجرا در محیط‌های خارج از مرورگرها نوشته شده‌اند، به آسانی به برنامه‌های وب تبدیل کرد.

WebAssembly به عنوان یک فناوری نوین در حوزه وب، کاربردهای بسیاری دارد. برای مثال، می‌توان در برنامه‌های وبی که برای پردازش تصویر، صوت و ویدئو استفاده می‌شوند، از WebAssembly استفاده کرد تا سرعت و عملکرد برنامه‌ها بهبود یابد. همچنین، با استفاده از WebAssembly می‌توان برنامه‌های بازی، برنامه‌های تحلیل داده، برنامه‌های کاربردی و ... را به صورت بهینه در مرورگرهای وب اجرا کرد.

## نحوه پیاده‌سازی WebAssembly

WebAssembly به صورت مستقل از هر مرورگری که استفاده می‌شود، عمل می‌کند. برای پیاده‌سازی WebAssembly، می‌توان از چندین روش استفاده کرد. این روش‌ها عبارتند از:

- استفاده از ابزارهای کامپایلر: برای تولید کدهای WebAssembly می‌توان از ابزارهای کامپایلر مختلفی مانند LLVM، Emscripten و ... استفاده کرد. با استفاده از این ابزارها، کدهایی که به زبان‌هایی مانند C، C++ و Rust نوشته شده‌اند، به کدهای WebAssembly تبدیل می‌شوند.
- استفاده از زبان‌های برنامه‌نویسی: برای نوشتن کدهای WebAssembly می‌توان از زبان‌هایی مانند AssemblyScript استفاده کرد. این زبان‌ها به برنامه‌نویسان اجازه می‌دهند تا کدهای WebAssembly را به شکل ساده‌تر و با سطح بالاتری از انتزاع نوشته و پیاده‌سازی کنند.
- استفاده از ویرایشگرها: برای پیاده‌سازی کدهای WebAssembly، می‌توان از ویرایشگرهایی مانند Visual Studio Code و Sublime Text استفاده کرد. این ویرایشگرها ابزارهایی را برای نوشتن، تست و اجرای کدهای WebAssembly فراهم می‌کنند.

## مثال‌های کد به زبان WebAssembly

برای نمونه، کد زیر یک عملیات جمع ساده را با استفاده از WebAssembly پیاده‌سازی می‌کند.

```
(module
  (func $add (param $a i32) (param $b i32) (result i32)
    get_local $a
    get_local $b
    i32.add)
  (export "add" (func $add)))
```

این کد یک ماژول WebAssembly را تعریف می‌کند که یک تابع با نام "add" دارد. این تابع دو پارامتر 32بیتی به نام "a" و "b" را می‌گیرد و جمع آن‌ها را به عنوان نتیجه برمی‌گرداند.

برای استفاده از این کد در یک برنامه وب، می‌توان از زبان‌هایی مانند JavaScript استفاده کرد. برای مثال، کد زیر نحوه صدا زدن تابع "add" از طریق WebAssembly را نشان می‌دهد.

```
fetch('add.wasm')
  .then(response => response.arrayBuffer())
  .then(buffer => WebAssembly.instantiate(buffer))
  .then(({instance}) => {
    const add = instance.exports.add;
    console.log(add(2, 3)); // output: 5
  });
```

این کد ابتدا فایل wasm را از سرور دریافت می‌کند، سپس آن را به صورت آرایه‌ای از بایت‌ها تبدیل می‌کند و با استفاده از WebAssembly.instantiate آن را به یک ماژول WebAssembly تبدیل می‌کند. سپس با استفاده از instance.exports، تابع "add" را به عنوان یک تابع JavaScript در دسترس قرار می‌دهد. در نهایت، تابع "add" با پارامترهای 2 و 3 فراخوانی شده و نتیجه آن در کنسول نمایش داده می‌شود.

## ویژگی‌های WebAssembly

WebAssembly دارای ویژگی‌هایی است که آن را به یک فناوری قدرتمند در حوزه وب تبدیل کرده است. این ویژگی‌ها عبارتند از:

- **سرعت بالا:** WebAssembly برای اجرای کدهایی که در زبان‌هایی مانند C، C++، Rust و ... نوشته شده‌اند بهینه شده است و به برنامه‌های وب اجازه می‌دهد تا با سرعت بالا اجرا شوند.

- **پرتابلیتی:** WebAssembly به عنوان یک فرمت اجرایی مستقل از مرورگر، قابلیت پرتابل بودن را دارد و به برنامه‌های وب اجازه می‌دهد تا بدون توجه به مرورگری که استفاده می‌شود، به صورت یکسان اجرا شوند.
- **امنیت:** WebAssembly با استفاده از یک مدل امنیتی محافظتی، به برنامه‌های وب اجازه می‌دهد تا در محیط امنی اجرا شوند و از تهدیدهای امنیتی مختلفی مانند حملات XSS و CSRF محافظت شوند.
- **قابلیت استفاده با زبان‌های مختلف:** WebAssembly به برنامه‌نویسان اجازه می‌دهد تا از زبان‌های مختلفی مانند C، C++، Rust و ... برای نوشتن کدهای خود استفاده کنند.
- **پشتیبانی از ابزارهای کامپایلر:** WebAssembly با پشتیبانی از ابزارهای کامپایلر مختلفی مانند LLVM، به برنامه‌نویسان اجازه می‌دهد تا به راحتی کدهای خود را به کدهای WebAssembly تبدیل کنند.
- **قابلیت اجرای برنامه‌های خارج از محیط وب:** با استفاده از WebAssembly، برنامه‌هایی که قبلاً برای اجرا در محیط‌های خارج از مرورگرها نوشته شده‌اند، به آسانی به برنامه‌های وب تبدیل کرد.

## ذکر چند مثال ساده

- تبدیل متن به عدد در WebAssembly:

این مثال یک تابع در C به نام `str_to_int` تعریف می‌کند که یک رشته را به عنوان ورودی گرفته و آن را به عدد تبدیل می‌کند.

```
int str_to_int(char* str) {
    int result = 0;
    while (*str) {
```

```

    result = result * 10 + (*str - '0');

    str++;
}

return result;
}

```

کد WebAssembly معادل این تابع به صورت زیر است:

```

(module
  (func $str_to_int (param $str i32) (result i32)
    (local $result i32)
    (local $char i32)
    (i32.const 0)
    set_local $result
    (block
      (loop
        (br_if 1 (i32.eqz (i32.load8_u (get_local $str))))
        (i32.const 10)
        mul
        (i32.load8_u (get_local $str))
        i32.const 48
        sub
        add
        set_local $char
        (get_local $result)
        (i32.const 10)
        mul
        (get_local $char)
        add

```

```

    set_local $result
    (i32.const 1)
    (get_local $str)
    i32.add
    set_local $str
    br 0
  )
)
(get_local $result)
)
(export "str_to_int" (func $str_to_int)))

```

- مثال ترسیم مستطیل در WebAssembly:

این مثال یک تابع در WebAssembly تعریف می‌کند که یک مستطیل با طول و عرض دلخواه را در یک کانواس به نام ctx ترسیم می‌کند.

```

(module
  (func $str_to_int (param $str i32) (result i32)
    (local $result i32)
    (local $char i32)
    (i32.const 0)
    set_local $result
    (block
      (loop
        (br_if 1 (i32.eqz (i32.load8_u (get_local $str))))
        (i32.const 10)
        mul
        (i32.load8_u (get_local $str))

```

```

i32.const 48
sub
add
set_local $char
(get_local $result)
(i32.const 10)
mul
(get_local $char)
add
set_local $result
(i32.const 1)
(get_local $str)
i32.add
set_local $str
br 0
)
)
(get_local $result)
)
(exports "str_to_int" (func $str_to_int)))

```

این تابع از دو تابع ایمپورت شده با نام `getContext` و `fillRect` استفاده می‌کند که در اینجا به تابع‌هایی در محیط جاوااسکریپت اشاره دارد. با استفاده از `getContext`، کانواسی با نام `ctx` به عنوان ورودی تابع گرفته می‌شود و با استفاده از `fillRect`، مستطیلی با طول، عرض، ارتفاع و موقعیت دلخواه در کانواس ترسیم می‌شود.

نتیجه‌گیری

WebAssembly به عنوان یک فناوری نوین در حوزه وب، به برنامه‌های وب اجازه می‌دهد تا با سرعت و کارایی بالاتری اجرا شوند. این فناوری با دارا بودن ویژگی‌هایی مانند سرعت بالا، پرتابلیتی، امنیت، قابلیت استفاده با زبان‌های مختلف و ... به برنامه‌نویسان اجازه می‌دهد تا به راحتی برنامه‌های خود را به شکلی بهینه در مرورگرهای وب اجرا کنند. با توجه به این ویژگی‌ها، WebAssembly در حال حاضر در بسیاری از برنامه‌های وب استفاده می‌شود و در آینده نیز احتمالاً بازیگر اصلی در حوزه توسعه برنامه‌های وب خواهد بود.