

IGALIA'S FOCUS AND GOALS 2020

WebKit Contributors Meeting 2019, Santa Clara

zdobersek@igalia.com, clima@igalia.com, rbuis@igalia.com



WHO WE ARE

- Worker cooperative with people all over the world
- Browsers is the main area of work (graphics, multimedia, etc.)
- We advocate for open source, open standards and secure platforms/software
- We actively participate in spec definition organizations (e.g. TC39 and W3C)

- We use WebKit to primarily enable integration of Web content into a wide range of use cases
- We are invested in helping WebKit remain competitive in those and other cases
- We support companies in ways that improve WebKit
 - Companies that would use WebKit
 - Companies that want to progress the Web platform

PORTS

THE GTK PORT

- Focuses on the Linux desktop.
- The main use cases are GNOME Web (Epiphany) and applications embedding web views.

THE WPE PORT

- Widely used in embedded (mainly Linux for now).
- Designed with performance, small footprint, accelerated content rendering, and simplicity of embedding in mind

THE WPE PORT

- Primary success in STB deployments
- Also used in different embedding use-cases: home appliances, in-vehicle information systems, digital signage
- Alternative to Chromium

API LAYER

- Solid PSON support
- GTK4 integration for WebKitGTK
- QtWPE
 - QML wrapper around WPE
- API additions as required

GRAPHICS

- Async scrolling support
 - Needs polish, more code-sharing
- Graphics pipeline revamp
 - Ongoing, loosening dependence on Cairo

- WebGL, WebGL2 -- ANGLE-based

Likely requires new ANGLE platform

- WebGPU

Depends on Vulkan support (result of graphics pipeline work)

MULTIMEDIA

GENERAL

- GStreamer playbin3 support
 - Improved seek support, stability
- GstWPE
 - Web content piped to GStreamer-managed pipeline
- RFC: GStreamer-specific tab in Web Inspector

WEBRTC

- Currently using libwebrtc
- Problems with hardware decoding integration
- Evaluating gstwebrtc
 - A more natural fit for our stack

MSE

- Platform support
Big rewrite pending to land

EME

- Platform support
 - Trying to find a good CDM integration solution
- SVP support

NETWORKING

- libsoup and glib-networking maintainance
GLib-based networking stack
- Client certificate authentication and PKCS #11 support
- HTTP2 and eventually QUIC+HTTP3

RELEASES & DEPENDENCIES

- One stable version every six months (in March and September)
- We branch for the new stable release around one month before the release
- Only bug fixes and improvements are allowed in stable branches, no new API or features

- We make several "unstable" releases from trunk during the cycle
 - After branching such releases are release candidates
- Additional releases from the stable branch after the initial stable release
 - Bug fixes, improvements and security issues

COMPILER UPDATES

- Debian and Ubuntu LTS releases support:
(GCC 7 OR Clang 6) AND libstdc++ 7, Ubuntu 18.04 (until April 2022)
(GCC 8 OR Clang 7) AND libstdc++ 8, Debian Buster (until late 2022)
- <https://trac.webkit.org/wiki/WebKitGTK/Dependencies>
- <https://trac.webkit.org/wiki/WebKitGTK/GCCRequirements>

WPE COMPILERS

- Build systems for Linux-based embedded devices often lag even further behind
- People targeting embedded devices are extremely reluctant to update their system compiler in the middle of product development
- In many cases future product updates will keep using the same compiler
- Expensive but usable work-arounds

QA & DEVELOPMENT

QA: TESTING INFRASTRUCTURE

- We provide and maintain the GTK/WPE/JSCOnly buildbots and EWS:
 - GTK: x86_64 testers. Performance bot
 - WPE: x86_64 testers
 - JSCOnly: ARM64/ARMv7/MIPS testers

- Plans to add more bots to the testing infrastructure:
 - Address, thread, undefined behavior sanitizers
 - WPE testers in ARMv7/ARM64 architecture
 - WPE performance bot
 - GTK API tester EWS

DEVELOPMENT

- Third-party dependency management is hard
- Current solution: JHBuild
- Future solution: Flatpak
 - Can be used as a development environment as well as a release channel

FASTER-PACE RELEASE CHANNELS

- Epiphany Tech Preview
<https://wiki.gnome.org/Apps/Web/Development>
Flatpak-based
- Simpler WPE consumption
Also something Flatpak-based
Pre-built images for popular embedded hardware

JSC

JSC 32-BIT SUPPORT

- LLInt, baseline JIT and DFG for ARMv7 and MIPS
- Enable as many tiers as possible
- Bug fixes
- EWS/buildbot
- Memory and performance work

JSC FEATURE IMPLEMENTATION

- Class features implementation
 - Static/instance class fields
 - Static/instance private methods

PUBLIC CLASS FEATURE EXAMPLE

```
class C {  
    field = 1;  
    static staticField = 2;  
}  
  
let o = new C();  
o.field; // returns 1  
  
C.staticField; // returns 2
```

PRIVATE CLASS FEATURE EXAMPLE

```
class C {  
  #field = 1;  
  static #staticField = 2;  
  #method() { return this.#field; }  
  
  static #staticMethod() { return this.#staticField; }  
  
  access() {  
    this.#method(); // returns 1  
    C.#staticMethod(); // returns 2  
  }  
}
```

JSC FEATURE IMPLEMENTATION

- BigInt (not feature complete)
 - Missing TypedArray support
 - Support '++' and '--' operators (including JIT)

TEST262 CONTRIBUTIONS

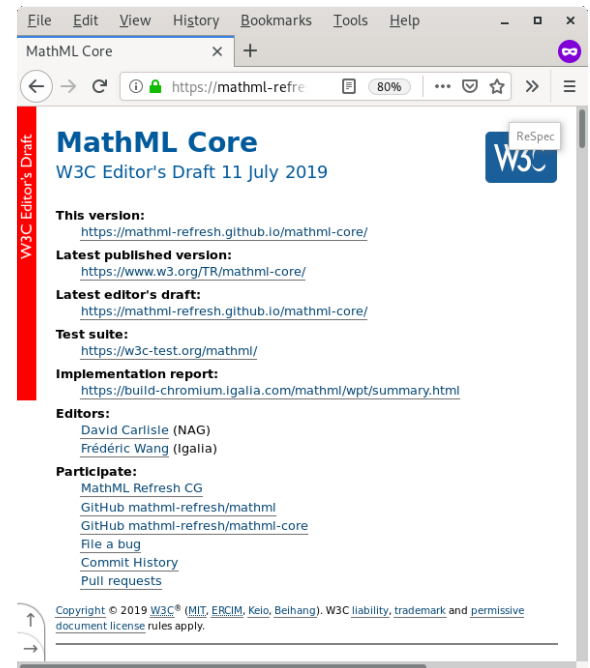
- Private methods and class fields
- BigInt

WEB STANDARDS

- Will not go into much detail for every 2020 Web Standard task.
- Feel free to ask about details.
- Listed responsible persons in the slides.
- The plans are not final yet.

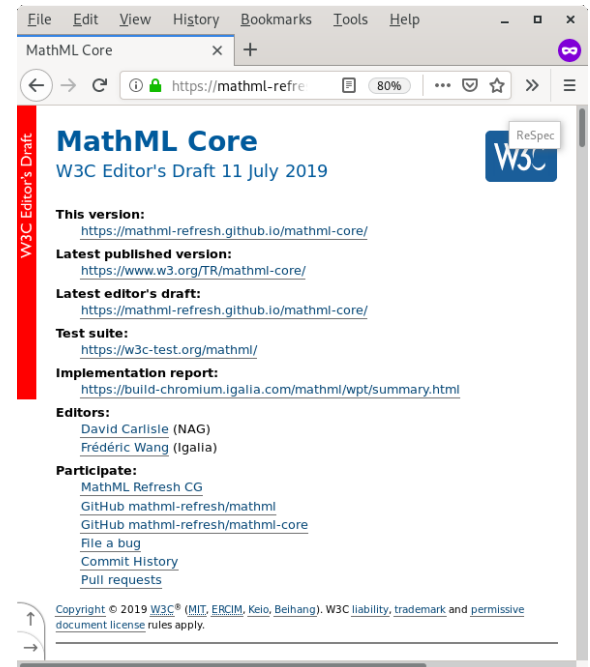
MATHML

- Specification
 - Core subset
 - TeX/OpenType
 - Web Platform



MATHML

- Specification
 - Core subset
 - TeX/OpenType
 - Web Platform
- WebKit
 - DOM/IDL
 - Layout
 - Cleanup



SCROLLING

- [scroll-behavior](#) (cchen, fwang)

SCROLLING

- [scroll-behavior](#) (cchen, fwang)
- [overscroll-behavior](#) (cchen, fwang)

SCROLLING

- [scroll-behavior](#) (cchen, fwang)
- [overscroll-behavior](#) (cchen, fwang)
- [scrollend/overscroll events](#) (cchen, fwang)

SCROLLING

- [scroll-behavior](#) (cchen, fwang)
- [overscroll-behavior](#) (cchen, fwang)
- [scrollend/overscroll events](#) (cchen, fwang)
- [ScrollIntoView](#) (cchen, fwang)

SCROLLING

- [scroll-behavior](#) (cchen, fwang)
- [overscroll-behavior](#) (cchen, fwang)
- [scrollend/overscroll events](#) (cchen, fwang)
- [ScrollIntoView](#) (cchen, fwang)
- [Scroll-To-Text?](#)

CSS

- Containment? (rego)

CSS

- Containment? (rego)
- CSS Grid maintenance (oriol)

CSS

- Containment? (rego)
- CSS Grid maintenance (oriol)
- Text Wrapping (from [Text Module](#)) (jfernandez)

CSS

- Containment? (rego)
- CSS Grid maintenance (oriol)
- Text Wrapping (from [Text Module](#)) (jfernandez)
- [Intrinsic Size](#) (cchen, fwang)

LOADING/NETWORK

- Lazy image + iframe loading (rbuis)

LOADING/NETWORK

- Lazy image + iframe loading (rbuis)
- Load prefetch/preload/prenavigate (rbuis)

LOADING/NETWORK

- Lazy image + iframe loading (rbuis)
- Load prefetch/preload/prenavigate (rbuis)
- Priority Hints (rbuis)

LOADING/NETWORK

- Lazy image + iframe loading (rbuis)
- Load prefetch/preload/prenavigate (rbuis)
- Priority Hints (rbuis)
- Stale-while-revalidate Cache-control directive (rbuis)

LOADING/NETWORK

- Lazy image + iframe loading (rbuis)
- Load prefetch/preload/prenavigate (rbuis)
- Priority Hints (rbuis)
- Stale-while-revalidate Cache-control directive (rbuis)
- Web Package (rbuis)
 - Signed HTTP Exchanges
 - Bundled HTTP Exchanges

OTHER STANDARDS

- OffscreenCanvas (clord)

OTHER STANDARDS

- OffscreenCanvas (clord)
- SVG (nzimmermann/WildFox)

OTHER STANDARDS

- OffscreenCanvas (clord)
- SVG (nzimmermann/WildFox)
- [Resize Observer](#) (cchen)

OTHER STANDARDS

- OffscreenCanvas (clord)
- SVG (nzimmermann/WildFox)
- [Resize Observer](#) (cchen)
- [Channel messaging](#)? (fwang)

INTEROPERABILITY

WEB PLATFORM TESTS

- Primary tool for
Conformance
Interoperability

WEB PLATFORM TESTS

- Primary tool for
Conformance
Interoperability
- Browsers daily tested at wpt.fyi

WEB PLATFORM TESTS

- Primary tool for
Conformance
Interoperability
- Browsers daily tested at wpt.fyi
[Safari Tech Preview](#)

WEB PLATFORM TESTS

- Primary tool for
Conformance
Interoperability
- Browsers daily tested at wpt.fyi
[Safari Tech Preview](#)
[WebKitGTK](#)

WEB PLATFORM TESTS

- Primary tool for
Conformance
Interoperability
- Browsers daily tested at wpt.fyi
[Safari Tech Preview](#)
[WebKitGTK](#)
- Infrastructure
Synchronization
Testing APIs

MAKING WEBKIT MORE INTEROPERABLE

- Identify test failures.

Bug triaging and gardening
Make them pass!

MAKING WEBKIT MORE INTEROPERABLE

- Identify test failures.
 - Bug triaging and gardening
 - Make them pass!
- Implement missing features.
 - Already imported tests
 - Client requests

MAKING WEBKIT MORE INTEROPERABLE

- Synchronize WPT tests.

Import existing tests

Export new or modified tests

MAKING WEBKIT MORE INTEROPERABLE

- Synchronize WPT tests.
 - Import existing tests
 - Export new or modified tests
- Work with the community
 - Standardization groups
 - Web developers / users

Thanks!



igalia