HI

ICANN85 मुंबई | CF - यह कैसे काम करता है: अंतर्राष्ट्रीयकृत डोमेन नाम
मंगलवार, 10 मार्च 2026 – 9:00 से 10:00 IST

चंपिका विजयतुंगा

सभी को फिर से नमस्कार। यह कैसे काम करता है - अंतर्राष्ट्रीयकृत डोमेन नाम में आपका स्वागत है। मेरा नाम चंपिका विजयतुंगा है और मैं इस सेशन के लिए रिमोट पार्टिसिपेशन मैनेजर हूँ। कृपया ध्यान दें कि यह सेशन रिकॉर्ड किया जा रहा है और यह ICANN समुदाय के प्रतिभागी आचार संहिता, ICANN के अपेक्षित व्यवहार मानक और ICANN की समुदाय विरोधी उत्पीड़न नीति के तहत संचालित है।

इस सत्र में भाग लेने के लिए कृपया नीचे दिए गए दिशा-निर्देशों का पालन करें। मैं इन्हें आपके संदर्भ के लिए चैट में भी पोस्ट कर दूंगी। इस सत्र के दौरान, सवालों को ज़ोर से तभी पढ़ा जाएगा अगर उन्हें सवाल-जवाब पॉड में सबमिट किया गया हो। इस सत्र के लिए अनुवाद में अंग्रेज़ी और हिन्दी भाषाएँ शामिल होंगी। अगर आप इस सत्र के दौरान बोलना चाहते हैं, तो कृपया अपना हाथ उठाएँ। कार्यक्रम स्थल पर मौजूद प्रतिभागी बोलने के लिए वास्तविक माइक्रोफ़ोन का उपयोग करेंगे।

अगर समय मिलता है, तो सेशन के दौरान केवल सवाल-जवाब सेक्शन में पूछे गए सवाल ही पढ़े जाएंगे। कृपया रिकॉर्ड के लिए अपना नाम और यदि आप

नोट: ऑडियो फ़ाइल को word/text डॉक्यूमेंट में ट्रांसक्राइब करने से आगे दिया गया परिणाम मिलता है। हालांकि ट्रांसक्रिप्शन काफ़ी हद तक सटीक है, फिर भी कुछ मामलों में सुनाई नहीं देने वाले पैसेज और व्याकरण संबंधी गड़बड़ियों के कारण वह अधूरा या गलत हो सकता है। उसे मूल ऑडियो फ़ाइल की सहायक के रूप में पोस्ट किया गया है, लेकिन उसे आधिकारिक रिकॉर्ड नहीं माना जाना चाहिए।

अंग्रेज़ी के अलावा कोई अन्य भाषा बोल रहे हैं तो उस भाषा का उल्लेख करें और मध्यम गति से साफ़-साफ़ बोलें। अब, मैं पिटिनन को मंच का जिम्मा सौंपता हूँ। धन्यवाद।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, चंपिका और सेशन में आप सभी का स्वागत है। तो, मेरा नाम है पिटिनन कूआर्मोर्नपटाना और मैं ICANN की IDN टीम से हूँ। और आज हम आपके साथ कुछ बिंदुओं पर चर्चा करेंगे। सबसे पहले, हम IDN या अंतर्राष्ट्रीयकृत डोमेन नामों के बारे में संक्षिप्त जानकारी प्राप्त करेंगे। और फिर हम आपको पर्दे के पीछे से यह भी दिखाएंगे कि इन सभी नियमों को तकनीकी रूप से कैसे डॉक्यूमेंट किया जाता है। और फिर हम कनाडा से एक समुदाय सदस्य को भी आमंत्रित करेंगे ताकि वे मूल निवासियों की भाषाओं पर अपने काम के बारे में बता सकें।

और अंत में, हमारे पास इंडोनेशिया से दूरस्थ वक्ता भी हैं जो अपनी प्रगति साझा करेंगे। तो, हमारा कार्यक्रम काफी व्यस्त है, इसलिए आप चाहें तो अपना सवाल यहां लिख सकते हैं, लेकिन हम सेशन के आखिर में ही उन पर विचार करेंगे। तो, सबसे पहले, यह समझने के लिए कि स्थानीय भाषाओं और लिपियों में IDN को कैसे सक्षम किया जाए। तो, अब हम IDN के बारे में बात करेंगे, कि यह कैसा दिखता है।

तो, आप ऊपर देख सकते हैं, ये अंतर्राष्ट्रीयकृत डोमेन नाम हैं, जो कि देवनागरी लिपि के लिए है, parathpasha.parath। और दरअसल, हमने सेशन्स पेज पर एक स्लाइड भी पोस्ट की है। तो, अगर आप स्लाइड डाउनलोड करते हैं, तो आप इस लिंक को कॉपी और पेस्ट कर सकते हैं, और यह वास्तव में ब्राउज़र पर काम करता है। तो, IDN फॉर्म के रूप में यह सबसे ऊपर है। हम इनमें से प्रत्येक को यूनिकोड वर्ण कहते हैं या U लेबल नाम से भी पुकारते हैं।

प्रोटोकॉल स्तर में परिवर्तित होने पर, इसे पर्दे के पीछे ASCII लेबल में परिवर्तित किया जाता है, जो X और डैश से शुरू होता है, जैसा कि आप देख रहे हैं। इसलिए, हम इसे A लेबल कहते हैं। और इसे सक्षम करने का तरीका मानकों के एक समूह से आता है जिसे हम IDNA2008 कहते हैं। यह IETF द्वारा विकसित एक मानक है और यह इस मानक ट्रैक और सूचनात्मक ट्रैक का एक सेट है।

इसमें इन सभी नामों की परिभाषाओं, प्रोटोकॉल, प्रोटोकॉल प्रॉपर्टी बनने के लिए यूनिकोड कोड पॉइंट श्रेणियों को कैसे प्राप्त किया जाए, और फिर अरबी या हिब्रू जैसी लिपि को कैसे संभाला जाए, इस बारे में बात की गई है, जो शायद दाएं से बाएं लिखी जाती है। तो, कुछ उदाहरण देने के लिए, एक RFC में, संख्या 5892 है, तो यह यूनिकोड कोड पॉइंट के गुण के बारे में है। अतः, अक्षरों की श्रेणी से व्युत्पन्न करके, उनके पास ये चार श्रेणियां हैं। सबसे पहले, प्रोटोकॉल

वैध है या नहीं, जिसे हम PVALID कहते हैं और यह आम तौर पर वह कोड पॉइंट होता है जिसका उपयोग DNS में किया जा सकता है।

उदाहरण के लिए, उच्चारण चिह्न वाला यह E, यूनिकोड कोड पॉइंट में 00E9 है। दूसरे प्रकार की विशेषता आवश्यक प्रासंगिक नियम हैं। तो, यह संभव है लेकिन इसके लिए कुछ शर्त होनी चाहिए। इसकी खुली छूट नहीं है। उदाहरण के लिए, लैटिन में मध्य बिंदु, माफ़ कीजिए, शायद लाल रंग की स्क्रीन पर इसे देखना थोड़ा मुश्किल है, 00B7 है। इसलिए, इस कोड पॉइंट का उपयोग केवल तभी किया जा सकता है जब यह दो L के बीच में आता हो। यह तब होने वाली उलझन को रोकने के लिए है, जब दूसरे स्तर और शीर्ष स्तर को अलग करने के लिए डॉट का उपयोग किया जाता है। और एक प्रकार ऐसा भी है जिसमें इसकी अनुमति नहीं है।

उदाहरण के लिए, यहाँ जो चिह्न और प्रतीक हैं जैसे प्रश्नवाचक चिह्न या हैश चिह्न, कभी-कभी प्रोटोकॉल इन्हें लेकर भ्रमित हो सकता है कि यह सिंटैक्स है या लेबल। और फिर अंतिम प्रकार का चिह्न NS चिह्न है। इसलिए प्रत्येक स्क्रिप्ट के लिए यूनिकोड कोड चार्ट में कुछ कोड पॉइंट्स पहले से ब्लॉक कर दिए जाते हैं, ताकि अगर कोई अतिरिक्त आवश्यकता हो या कोई बदलाव करना पड़े, तो उसके लिए जगह उपलब्ध रहे। इसलिए, उनमें से कुछ का उपयोग नहीं किया जा सकता है।

इसलिए, ये वो कोड पॉइंट हैं जिन्हें असाइन नहीं किया जा रहा है। इन सब बातों के बावजूद, मानक में यह कहा गया है कि अगर कोई चीज़ PVALID है, तो इसका मतलब यह नहीं है कि उसे सीधे DNS में उपयोग किया जा सकता है, क्योंकि इसके अलावा भी कुछ वैरिएंट परिभाषाएँ या अन्य नियम हो सकते हैं जिन्हें ध्यान में रखना ज़रूरी है। और इसी कारण हमें इन नियमों को बनाना पड़ता है, जिन्हें लेबल जेनरेशन रूल्स कहा जाता है, जिन्हें हम इस सेशन में दिखाएंगे कि ये कैसे काम करते हैं।

लेबल जेनरेशन रूल्स का सीधा-सीधा मतलब है किसी विशेष स्क्रिप्ट या भाषा में लेबल बनाने के लिए कौन-कौन से नियम लागू होते हैं और उन्हें कैसे इस्तेमाल किया जाए। इसके चार भाग हैं। मेटाडेटा इन नियमों को समझाने का काम करता है। इसमें ये सारी जानकारी शामिल होती है जैसे वर्ज़न, तारीख, भाषा और आवश्यकता अनुसार आप एक विवरण भी जोड़ सकते हैं। इसके बाद इसमें तीन मुख्य भाग होते हैं—कोड पॉइंट रिपर्टॉयर, वैरिएंट, और कॉन्टेक्स्चुअल रूल्स। इसलिए, हम एक-एक करके प्रत्येक सेक्शन में जाते हैं। लेकिन उससे पहले, कुछ डिजाइन सिद्धांतों पर भी विचार करना आवश्यक है।

हमारे पास RFC 6912 है, जो यह निर्धारित करता है कि कौन-से यूनिकोड कोड पॉइंट्स लेबल में शामिल किए जा सकते हैं और DNS में उपयोग किए जा सकते हैं। यहाँ कई सिद्धांत हैं, उदाहरण के लिए दीर्घकालिकता, जब यूनिकोड में नए

अक्षर जोड़े जाते हैं, तो कई बार ऐसा होता है कि अगर उन्हें बाद के वर्ज़नों में नए सिरे से एन्कोड किया गया है, तो सॉफ़्टवेयर या डिवाइस इस नए मानक के साथ तालमेल न बिठा पाएं।

भले ही वह अक्षर पहले से ही यूनिकोड में एन्कोड किया गया हो, लेकिन हो सकता है कि कीबोर्ड में वह अक्षर मौजूद न हो। इसलिए हम सुझाव देते हैं कि डोमेन नाम में उपयोग होने वाले कोड पॉइंट्स यूनिकोड के ऐसे वर्ज़न में हों जो काफी समय से प्रचलन में हों, ताकि उनकी स्थिरता सुनिश्चित हो सके। डिज़ाइन के हिसाब से, यह सबसे कम आश्चर्यजनक भी होना चाहिए।

उदाहरण के लिए, अगर आप किसी भाषा या स्क्रिप्ट का उपयोग करते हैं और आपने उस अक्षर को अपने रोज़मर्रा के जीवन में कभी भी उपयोग होते हुए नहीं देखा है, लेकिन वह अचानक डोमेन नाम में दिखाई दे, तो यह भ्रम पैदा कर सकता है और हो सकता है कि उस स्क्रिप्ट को इस्तेमाल करने वाले लोग उसे समझ न पाएं। यह प्रासंगिक सुरक्षा भी होनी चाहिए। इसलिए वे कोड पॉइंट्स जिनके इस्तेमाल के लिए कुछ संदर्भ की ज़रूरत होती है, वे संदर्भ मौजूद होने चाहिए।

इसके अलावा, रूढ़िवादिता का सिद्धांत भी है; ऐसा इसलिए है क्योंकि डोमेन नाम एक साझा क्षेत्र होता है, और मूल रूप से लोग दुनिया भर में एक साथ मिलकर उस नाम के संसाधन का उपयोग कर सकते हैं। एक सिद्धांत यह भी है कि जब आपको यह साफ़ न हो कि किसी कोड पॉइंट का इस्तेमाल किया

जाना चाहिए या नहीं, तो हमेशा ज़्यादा सुरक्षित और महफ़ूज़ तरीका अपनाना बेहतर होता है जो कि ज़्यादा सतर्कता भरा होता है। अगला सिद्धांत समावेशन है।

इसलिए, यह तय करने के लिए कि किस कोड पॉइंट का उपयोग किया जाना चाहिए, वास्तव में इसकी शुरुआत एक खाली सेट से होनी चाहिए, और फिर एक-एक करके वर्णों का विश्लेषण करना शुरू करना चाहिए। और यदि इसका उपयोग करना वैध है, तो आप इसे धीरे-धीरे एक-एक करके जोड़ते जाते हैं। यह शुरुआत इस तरह नहीं होनी चाहिए कि पहले सब कुछ शामिल कर लिया जाए और बाद में जो ज़रूरी नहीं है उसे हटाया जाए। इसलिए प्रक्रिया एक खाली सेट से शुरू होनी चाहिए और फिर धीरे-धीरे शामिल किया जाना चाहिए।

ये सभी कुछ प्रमुख सिद्धांत हैं जिन्हें मैं विशेष रूप से बताना चाहता/चाहती हूँ। लेकिन अंत में स्थिरता का सिद्धांत बहुत महत्वपूर्ण है, क्योंकि अगर आप किसी चीज़ को डोमेन नाम में उपयोग की अनुमति दे देते हैं, तो बाद में उसे हटाना बहुत मुश्किल हो जाता है। इसलिए किसी को उपयोग की अनुमति देने से पहले इन सभी सिद्धांतों का सावधानीपूर्वक विश्लेषण करना ज़रूरी है। इसलिए इसके पहला भाग कोड पॉइंट रिपर्टॉयर में हमें समुदाय के साथ मिलकर यह तय करना होता है कि उस विशेष स्क्रिप्ट के लिए कौन-से कोड पॉइंट्स उपयोग किए जाएँ।

डिज़ाइन का मुख्य उद्देश्य, जैसा कि याद दिलाना चाहूँगा, यह है कि डोमेन नाम मूल रूप से एक स्मरणीय प्रणाली है। और यह एक ऐसा माध्यम है, जिससे हम इंसान IP एड्रेस को याद रखने के बजाय आसानी से नाम याद रख सकते हैं। इसलिए इन सभी डिज़ाइन को सुरक्षित और स्थिर बनाए रखना आवश्यक है। अगर समाधान ही भ्रम पैदा करता है, तो यह उद्देश्य पूरा नहीं हो पाएगा।

और क्योंकि यह सिर्फ एक नाम है, इसलिए इसमें किसी भाषा या स्क्रिप्ट, व्याकरण या भाषाई आवश्यकताओं को पूरी तरह कवर करना ज़रूरी नहीं होता। यह जरूरी नहीं कि कोई वास्तविक शब्द ही हो। उदाहरण के लिए, यह संक्षिप्त रूप हो सकते हैं। और यह ज़रूरी नहीं है कि वर्तनी बिल्कुल सही हो, लेकिन यह स्थिर और अनुमानित होनी चाहिए। तो चलिए, ASCII डोमेन नाम के बारे में थोड़ा पीछे चलते हैं। इसलिए, ASCII में भी कुछ ऐसे नियम हैं जिनके बारे में हमें शायद जानकारी नहीं है।

इसलिए, ASCII अक्षरों के शीर्ष स्तर पर, आप केवल A से Z तक के अक्षरों का ही उपयोग कर सकते हैं। ऐसा इसलिए है क्योंकि अगर आप शीर्ष स्तर पर भी संख्याओं का उपयोग करते हैं, तो IP एड्रेस को लेकर भ्रम की स्थिति उत्पन्न हो सकती है। इसलिए, शीर्ष स्तर के मानक के अनुसार, आप केवल अक्षर का उपयोग कर सकते हैं। लेकिन दूसरे या तीसरे स्तर पर, जाहिर है आप अक्षर, अंक और हाइफ़न का उपयोग कर सकते हैं। कुछ नियम पहले से ही लागू हैं। और यह ASCII के लिए कोड चार्ट है।

इसलिए ASCII मूल रूप से वे अक्षर हैं जो आप आमतौर पर कीबोर्ड पर देखते हैं। तो, अक्षरों के अलावा, आपको कुछ चिह्न भी दिखाई देते हैं, जैसे गणितीय चिह्न और कुछ कीबोर्ड नियंत्रण चिह्न, जैसे स्पेस, नई लाइन इत्यादि। इसलिए, अगर यह शीर्ष स्तर का है, तो केवल हरे रंग में हाइलाइट किए गए अक्षरों का ही उपयोग किया जा सकता है। लेकिन अगर यह दूसरा स्तर है, तो आप अक्षरों, अंकों और फिर हाइफ़न का उपयोग कर सकते हैं। इसलिए ऐसे अक्षरों को बाहर कर देना चाहिए।

जब हम यूनिकोड की बात करते हैं, तो यह एक अलग संस्था यूनिकोड कंसोर्टियम द्वारा संचालित होता है, जो विभिन्न स्क्रिप्ट के अक्षरों को 4 या 5 अंकों के हेक्साडेसिमल नंबर में एन्कोड करता है, जिन्हें हम यूनिकोड कोड पॉइंट कहते हैं, और इन्हें एक चार्ट के रूप में प्रस्तुत किया जाता है। तो, बाईं ओर अरबी चार्ट का उदाहरण है, और दाईं ओर इथियोपियाई चार्ट का आंशिक उदाहरण है। अभी इन सभी चीज़ों के लिए यूनिकोड का वर्ज़न 17 चल रहा है।

इसमें अब तक 172 स्क्रिप्ट्स को एन्कोड किया जा चुका है। कल्पना कीजिए कि अगर हम बाएँ तरफ से शुरू करें, जो कि ASCII स्पेस है, तो इसे हम एक तरह से एक चार्ट या एक स्क्रिप्ट की शुरुआत मान सकते हैं। उपयोग के लिए केवल 63 कोड पॉइंट ही चुने गए हैं, और इससे शायद I और L, o और शून्य आदि के बीच कुछ भ्रम पैदा हो सकता है। लेकिन अब अगर हम इसे यूनिकोड के

अनुसार सक्षम कर दें, जहाँ 172 स्क्रिप्ट्स और लगभग 300,000 अक्षर हैं, तो भ्रम की संभावना और भी बढ़ सकती है।

इसलिए हमें इसे सावधानीपूर्वक तरीके से सीमित करना होता है और समुदाय के साथ मिलकर यह तय करना पड़ता है कि उस स्क्रिप्ट में नामों को कवर करने के लिए न्यूनतम कितने कोड पॉइंट्स वास्तव में आवश्यक हैं। और इस तरह हम कोड पॉइंट का निर्धारण करते हैं। इस पर काम करने के लिए हम उस स्क्रिप्ट के कोड चार्ट की समीक्षा से शुरुआत करते हैं और फिर सबसे बुनियादी सवाल से लेकर आगे तक यह तय करते हैं कि क्या यह प्रोटोकॉल के अनुसार वैध है या नहीं।

और फिर आगे बढ़ते हुए, क्या भाषा या स्क्रिप्ट में इसकी आवश्यकता है? क्या यह पहचानकर्ता के लिए उपयुक्त है? कभी-कभी इसका उपयोग रोजमर्रा की जिंदगी में होता है, लेकिन वास्तव में इसका उपयोग नामों में नहीं किया जाता है, जैसे कि दोहराए जाने वाले चिह्न इत्यादि। क्या वह यूनिकोड एन्कोडिंग पहले से ही स्थिर है? यह कितना सुलभ है? उदाहरण के लिए, क्या ये अक्षर इनपुट कीबोर्ड पर उपलब्ध हैं या नहीं? और फिर हमें हमेशा RFC 6912 पर वापस जाना पड़ता है। क्या ऐसा कोई सिद्धांत है जिसका उल्लंघन इस कोड पॉइंट के जुड़ने से हो सकता है?

यही कोड पॉइंट वाला भाग होता है। और अगले भाग में हम इसके कुछ उदाहरण भी देखेंगे। लेकिन अभी के लिए, चलिए दूसरे भाग, यानी वेरिएंट पर

चलते हैं। IDN की एक और अवधारणा यह है कि हमारे पास कुछ कोड पॉइंट या कोड पॉइंट का समूह भी होता है, जिन्हें स्क्रिप्ट समुदाय समान मान सकता है। और चलिए मैं आपको कुछ उदाहरण दिखाता हूँ।

मुख्य रूप से दो प्रकार होते हैं। पहला वाला देखने में एक जैसा होता है, और आमतौर पर हम सुरक्षा कारणों से इन्हें परिभाषित करते हैं। अगर आप नीले रंग में लिखा टेक्स्ट देखते हैं ऊपर बाईं ओर 'EPIC', और दाईं ओर 'EPIC' जैसा दिखने वाला टेक्स्ट और अगर आप इसके पीछे का कोड पॉइंट देखते हैं, तो यह 0065 से शुरू हो रहा है, जबकि दूसरा वाला 0435 से शुरू हो रहा है। मूल रूप से, ये दो अलग-अलग स्क्रिप्ट हैं, क्षमा करें, दो अलग-अलग, हाँ, दो अलग-अलग स्क्रिप्ट और दो अलग-अलग लेबल।

लेकिन इंसानी आंखों के लिए, हम वास्तव में दोनों में अंतर नहीं बता सकते, ये लैटिन और सिरिलिक स्क्रिप्ट हैं। इसलिए, इस प्रकार के वेरिएंट में, हमें सुरक्षा उद्देश्य के लिए उन्हें वेरिएंट के रूप में परिभाषित करना होगा। और वास्तव में, लैटिन स्क्रिप्ट समुदाय और सिरिलिक स्क्रिप्ट समुदाय एक साथ आते हैं, मिलकर काम करते हैं, और परिभाषित करते हैं, जैसे कि ठीक है, 0065 और 0435 वेरिएंट। एक अन्य प्रकार का वेरिएंट भी है, जो उपयोगिता के लिए परिभाषित किया जाता है।

और यह मुख्य रूप से हान स्क्रिप्ट या चीनी स्क्रिप्ट तथा अरबी स्क्रिप्ट पर लागू होता है। इसलिए, चीनी स्क्रिप्ट के लिए, आप देख सकते हैं कि दो लेबल

हैं। उन लेबलों का दूसरा अक्षर थोड़ा अलग है क्योंकि यह सरलीकृत चीनी और पारंपरिक चीनी का मिश्रण है। इन दोनों संस्करणों का उपयोग विश्व के विभिन्न क्षेत्रों में किया गया है।

इसलिए, यदि आप चीन की मुख्य भूमि में काम करने वाले एक ब्रांड के मालिक हैं, लेकिन आप चाहते हैं कि आपका ब्रांड अन्य स्थानों से भी पहुंच सके, तो आप दोनों नामों को एक ही समय में सक्रिय रखना चाह सकते हैं। और इसी कारण चीनी समुदाय ने अंतिम कोड बिंदु को ई-चार्टर के एक प्रकार के रूप में परिभाषित किया। और ऐसा ही प्रभाव अरबी भाषा के मामले में भी हुआ है।

इसलिए, वेरिएंट को परिभाषित करने में भी कुछ ऐसे कार्य शामिल हैं, जिन्हें लेबल जनरेशन के नियम बनाने के लिए पूरा करना आवश्यक है। इसका उद्देश्य यह सुनिश्चित करना है कि निमोनिक प्रणाली भ्रम को न्यूनतम करे, अतः हमें इन्हें स्क्रिप्ट के भीतर और स्क्रिप्ट के बाहर दोनों ही जगहों पर एक समान रूप से परिभाषित करना होगा। और क्योंकि हम रूढ़िवादिता की आवश्यकता पर विचार कर रहे हैं, इसलिए ज्यादातर समय हम इसे सुरक्षा मुद्दों के लिए परिभाषित करते हैं, इसलिए हम इसे एक ब्लॉक वेरिएंट के रूप में परिभाषित करते हैं, जिसका अर्थ है कि यदि एक संस्करण पहले से मौजूद है, तो दूसरा संस्करण अब मौजूद नहीं होना चाहिए।

लेकिन कुछ न्यूनतम मामलों में, हम चाहते हैं कि वे एक ही समय में सक्रिय हों, और इसके लिए, हम इसे आवंटन योग्य प्रकार के रूप में परिभाषित करते

हैं। और यह कार्य भी कोई मामूली बात नहीं है क्योंकि कभी-कभी आपको समुदाय के भीतर कुछ सहमति बनानी पड़ती है और साथ ही निष्कर्ष का खंडन करने के लिए पूरे समुदाय के साथ मिलकर काम भी करना पड़ता है। आगे बढ़ते हुए, वैरिएंट को परिभाषित करने के लिए हमें यह सवाल पूछना होता है कि क्या ऐसे दो कोड पॉइंट्स हैं जिन्हें उस स्क्रिप्ट के मूल उपयोगकर्ता एक-दूसरे के स्थान पर इस्तेमाल करने योग्य या समान मानते हैं? और क्या उसके कोई वैकल्पिक रूप भी मौजूद हैं? और क्या किसी विशेष संदर्भ में कोई वैरिएंट मौजूद है? और क्या इन वैरिएंट्स का किसी अन्य सिद्धांतों के साथ कोई टकराव तो नहीं है?

ये सभी वैरिएंट से संबंधित भाग होते हैं। और अब चलिए अंतिम भाग, यानी नियमों से संबंधित भागों पर चलते हैं। कुछ स्क्रिप्ट्स में, जैसा कि RFC में बताया गया है, उनकी प्रकृति या कोड पॉइंट्स के कारण कुछ प्रतिबंध लगाने पड़ते हैं और सुरक्षित व स्थिर उपयोग सुनिश्चित करने के लिए संदर्भ आधारित नियमों की आवश्यकता होती है। और इसके लिए, फिर से ध्यान दिलाना है कि यह व्याकरण के नियमों से संबंधित नहीं है।

इसका उद्देश्य सिर्फ यह सुनिश्चित करना है कि स्क्रिप्ट का उपयोग इस तरह हो कि उपयोगकर्ता के लिए कुछ भी अप्रत्याशित या भ्रमित करने वाला न हो। सुरक्षा संबंधी समस्याओं के उत्पन्न होने की संभावना को कम करें, उपयोगिता संबंधी बाधाओं के उत्पन्न होने की संभावना को कम करें। और

कुछ मामलों में ये नियम इसीलिए बनाए जाते हैं ताकि आवंटित किए जा सकने वाले वैरिएंट लेबल्स की संख्या को कम किया जा सके। और चलिए मैं आपको कुछ उदाहरण दिखाता हूँ।

तो, ये उन नियमों का उदाहरण है जो जोखिम को कम करने के उद्देश्य से लागू किए गए हैं। इसलिए, अगर आप बाईं और दाईं ओर यह कोड पॉइंट देखें, तो यह दिखने में बहुत समान लगते हैं लगभग एक जैसे दिखने वाले अक्षर की तरह। यह ऐसा दिखता है जैसे एक सीधी खड़ी लाइन हो, जिसके नीचे थोड़ा सा सिरा मुड़ा हुआ हिस्सा हो। बाईं ओर, केवल एक कोड पॉइंट है, 0E41। लेकिन दाईं ओर जो दिख रहा है, वह असल में दो कोड पॉइंट्स का संयोजन है। एक सीधी खड़ी लाइन है, जिसका कोड पॉइंट 0E40 है।

इस स्थिति में, अगर आप इन अक्षरों को साथ-साथ न देखकर अलग से, जैसे ब्राउज़र में देखें, तो आपको लग सकता है कि दोनों बिल्कुल एक जैसे हैं। इस स्थिति में, TypeScript के नियमों में अनुसार, हमने यह निर्धारित किया है कि 0E40 जैसे कोड पॉइंट एक-दूसरे के बाद नहीं आ सकते, ताकि इस तरह का भ्रम पैदा करने वाला मामला न हो। इसमें स्थिरता संबंधी समस्याएं भी हैं।

उदाहरण के लिए, कुछ स्क्रिप्टस में आपको कुछ ऐसी संरचनात्मक चीज़ें मिल सकती हैं, जैसे कि टोन मार्क जो कि एक तरह का एक्सेंट होता है, इसका मूल नियम यह है कि एक व्यंजन पर केवल एक ही टोन मार्क होना चाहिए, क्योंकि आप एक ही समय में दो अलग-अलग सुर नहीं गा सकते। यह लाओ स्क्रिप्ट

से संबंधित है। अगर आप दो टोन मार्क्स को एक साथ टाइप करते हैं, तो कुछ सिस्टम उन्हें एक-दूसरे के ऊपर ही दिखा देते हैं ठीक वैसे ही, जैसा कि बाईं ओर Android फ़ोन पर दिख रहा है। लेकिन कुछ अन्य सिस्टम इसे पिछले वाले के बगल में प्रदर्शित कर सकते हैं, जैसे कि iPhone पर।

इस मामले में, इससे अस्थिरता और अनिश्चितता पैदा होती है, और उपयोगकर्ता भी हैरान रह जाते हैं क्योंकि फ़ॉन्ट और बाकी सभी चीज़ें इसे रेंडर करने के लिए तैयार नहीं होतीं। लाओ स्क्रिप्ट में इस तरह के मामलों में, कुछ ऐसे नियम भी हैं जो एक टोन मार्क के बाद दूसरे टोन मार्क के आने की अनुमति नहीं देते। और LGR के इस्तेमाल की बात करें, तो जब हमारे पास ये नियम लागू होते हैं चूँकि ये मशीन पठनीय होते हैं इन्हें टूल आसानी से समझ सकता है, और जब ये पहले से ही इंस्टॉल होते हैं, तो हम इनमें लेबल डाल सकते हैं; यह उन लेबल को मान्य कर सकता है और साथ ही उन लेबल के अलग-अलग रूप भी तैयार कर सकता है।

और मेरे हिस्से के आखिर में, हमारे पास दो प्रोजेक्ट हैं जो LGR में शामिल हैं। एक है रूट ज़ोन LGR, जो शीर्ष स्तर के लिए नियम है। इस प्रोजेक्ट के तहत, हम अलग-अलग स्क्रिप्ट कम्युनिटी के साथ काम कर रहे हैं जिन्हें हम जेनरेशन पैनल्स कहते हैं और फिर हमने उन्हें रूट ज़ोन LGR के एक कलेक्शन में इंटीग्रेट कर दिया है। अभी हम इसके 6वें वर्ज़न पर हैं, जिसमें पहले से ही 27

स्क्रिप्ट्स शामिल हैं जो लगभग 400 भाषाओं को सपोर्ट करती हैं। यह 45 देशों के 18 पीढ़ियों के पैनलों के साथ 11 वर्षों का काम है।

तो, 10,000 से अधिक घंटे स्वयंसेवा में बिताए गए। ये वे 27 स्क्रिप्ट्स हैं जिन्हें हम कवर करते हैं, और यही वह आधार भी है जिसके अनुसार यह तय होता है कि आने वाले राउंड में किन स्क्रिप्ट्स के लिए आवेदन किया जा सकता है। हमारा एक और प्रोजेक्ट दूसरे लेवल पर एक रेफ़रेंस है, इसकी ज़रूरत इसलिए पड़ती है, क्योंकि अगर आप कोई TLD चलाते हैं और अपने TLD के तहत कोई भाषा या स्क्रिप्ट उपलब्ध कराना चाहते हैं, तो आपको लेबल जनरेशन रूल बनाना होगा जिसे कभी-कभी IDN टेबल भी कहा जाता है और इन नियमों की समीक्षा ICANN द्वारा की जानी ज़रूरी है।

इसलिए, हमने उन टेबल्स की स्थिर तरीके से समीक्षा करने के लिए रेफ़रेंस LGR विकसित किया। और ये हमारे पास मौजूद संदर्भ LGR की सूची है। हमारे पास दोनों हैं — भाषा-आधारित और स्क्रिप्ट-आधारित। और आप इसे बाद में प्रजेंटेशन में देख सकते हैं, मैं अब आगे बढ़ता हूँ। इसलिए, इन सभी रेफ़रेंस LGR को विकसित करने के लिए हम वास्तव में अलग-अलग समुदायों के साथ, रिक्वेस्ट के आधार पर सहयोग कर रहे हैं।

और समुदाय से हमें जो स्वैच्छिक विशेषज्ञता चाहिए, वह स्पष्ट रूप से भाषाई विशेषज्ञों की है, जो रिपर्टॉयर, वेरिएंट और नियमों को विकसित करें। और साथ ही, हम समुदाय से शब्दों की सूची प्रदान करने का अनुरोध करेंगे, ताकि

हम आगे का विश्लेषण और जांच कर सकें। और ICANN की ओर से, हम उन पहचान विवरणों को नियमों में बदलेंगे और तकनीकी LGR प्रदान करेंगे और फिर समुदाय से उसकी समीक्षा करने के लिए कहेंगे।

अंत में, जब इसे समुदाय द्वारा समीक्षा कर लिया जाता है, तो ICANN इसे अंतिम रूप देने के लिए पब्लिक कमेंट में ले जाएगा और फिर इसे मौजूदा रेफ़रेंस LGR में एकीकृत करेगा। इसलिए, अगर आप अपनी अतिरिक्त लिपि भाषा को सक्षम करने में रुचि रखते हैं, तो कृपया हमसे संपर्क करें और हम आगे की प्रक्रिया शुरू कर सकते हैं।

तो अब, मैं इसे अक्षत जोशी को सौंपूंगा, जो पहले हमारे साथ रूट ज़ोन LGR प्रोजेक्ट्स में एक नियो-ब्राह्मी स्क्रिप्ट जनरेशन पैनल सदस्य के रूप में काम कर चुके हैं, और वे हमें यह समझाएंगे कि इसे तकनीकी भाग में कैसे एन्कोड किया जाए। अक्षत, अब आपकी बारी है।

अक्षत जोशी

धन्यवाद। रिकॉर्ड के लिए, मैं अक्षत बोल रहा हूँ। तो, हमने देखा है कि हमारे पास कितने भाषा LGR हैं और वे वास्तव में कैसे काम करते हैं और उनका उद्देश्य क्या है। इस सेक्शन में हम देखेंगे कि एक LGR वास्तव में कैसा दिखता है, इसे वास्तव में कैसे बनाया जाता है, और अगर आपकी तरफ कोई उपयोग

का मामला है, तो आप जिस भाषा को अपनाने का इरादा रखते हैं, उसके लिए LGR का उपयोग कैसे कर सकते हैं।

तो, जैसे कि पिटिनन पहले ही समझा चुकी हैं, इसमें चार सेक्शन होते हैं, जैसे मेटाडेटा सेक्शन। मैं बस इसका सारांश बता रहा हूँ क्योंकि अगली स्लाइड में हम इसे वास्तविक XML में देख पाएंगे। एक है मेटाडेटा सेक्शन, और दूसरा है नॉर्मेटिव सेक्शन, जो कोड पॉइंट को दर्शाने के लिए होता है। कोड पॉइंट्स के साथ ही वेरिएंट जानकारी भी जुड़ी होती है और अंत में हमारे पास तीन प्रकार के नियम होते हैं — कॉन्टेक्स्ट नियम, लेबल इवैल्यूएशन नियम, और एक्शन नियम।

ये LGR दो रूपों में आते हैं। एक पूरी तरह से मशीन-पठनीय फ़ॉर्मेट होता है, जो XML है और दूसरा इसका HTML वर्ज़न होता है, जिसे आप उस विशेष वर्ज़न को जनरेट करने के लिए LGR टूल का उपयोग करके प्राप्त कर सकते हैं। LGR टूल की बात करें, तो यह ICANN द्वारा प्रदान किया गया एक टूल है, जिसका उपयोग LGR को विकसित करने और प्रबंधित करने के लिए किया जाता है। इसके अतिरिक्त, अगर आपके पास कुछ लेबल्स का सेट है, जिन्हें आप किसी पहले से मौजूद LGR या आपके द्वारा बनाए गए LGR के माध्यम से चलाना चाहते हैं, तो आप इस टूल का उपयोग भी कर सकते हैं।

और अगर आप किसी रजिस्ट्री में से हैं और अपने पास मौजूद कुछ IDN टेबल्स को यह जांचना चाहते हैं कि वे किसी विशेष पहले से मौजूद LGR के अनुरूप हैं

या नहीं, तो आप इस टूल का उपयोग भी कर सकते हैं। इस टूल तक कैसे पहुंचा जा सकता है? आप lgrtool.icann.org पर जा सकते हैं और इसके लिए आपको ICANN का एक अकाउंट चाहिए होगा, जिसके बाद आप इस टूल का स्वतंत्र रूप से उपयोग शुरू कर सकते हैं। इस टूल के लिए एक पूरा विस्तृत यूज़र गाइड भी उपलब्ध है, जो स्लाइड में दिए गए लिंक पर मिल सकता है।

जो लोग इस टूल को इसके API के ज़रिये एक्सेस करना चाहते हैं, उनके लिए इस टूल का कोड भी ओपन सोर्स किया गया है और आप उस वर्ज़न का भी उपयोग कर सकते हैं। ठीक है, इस सेशन में LGR बनाने के उद्देश्य से, हम एक काल्पनिक भाषा का उदाहरण लेंगे — A से G भाषाएँ, यानी H-A से G भाषा। इस भाषा की क्या विशेषताएं हैं? इस भाषा में लगभग 11 लैटिन अक्षर और एक संकेत होता है।

ठीक है, यही वह है जिसका उपयोग A से G भाषा का समुदाय अपने दैनिक संचार में करता है। भाषा की ओर से कुछ अतिरिक्त आवश्यकताएँ भी हैं — यदि आप कोड पॉइंट्स को देखें, तो उसमें F अक्षर के दो प्रकार मौजूद हैं। एक F सामान्य है, दूसरा हुक के साथ है और समुदाय उन्हें एक ही मानता है। इसलिए, इससे एक वैकल्पिक विन्यास की आवश्यकता का आधार बनता है। और फिर एक एलोवेरा का क्लिक अक्षर है जो किसी भी लेबल की शुरुआत में कभी नहीं होना चाहिए। इस विशेष भाषा के लिए यही आवश्यकता है।

अब, अगर हम यूनिकोड कोड चार्ट को देखें, तो ये सभी अक्षर मूल रूप से लैटिन लिपि से आए हैं, लेकिन तीन अलग-अलग कोड चार्ट से। यह मूल लैटिन है, जिसमें A से G तक सभी अक्षर हैं। और फिर लैटिन-1 सप्लीमेंट में हमारे पास तीन अक्षर हैं — A पर एक्यूट, E पर एक्यूट, और एक डिग्री साइन भी मौजूद है, जो लाल बॉक्स में दिखाया गया है। इसके बाद, हमारे पास लैटिन एक्सटेंडेड B है — बस पिछली स्लाइड पर वापस जाएँ, पिटिनन। लैटिन एक्सटेंडेड B कोड चार्ट में हमारे पास हुक के साथ F है, और मेरा मानना है कि वह एक अलाउ-विलर क्लिक अक्षर है।

अब डिग्री के चिह्न के चारों ओर एक लाल वर्ग है और हम अगली स्लाइड में देखेंगे कि ऐसा क्यों है। तो, अब जब हमने देख लिया है कि किसी कोड पॉइंट को अपने संग्रह में शामिल करने के लिए किन-किन प्रश्नों से गुजरना पड़ता है। और इन प्रत्येक अक्षरों के लिए, जब हम इन सभी प्रश्नों से गुजरते हैं, तो लगभग सभी पूरी तरह सही निकलते हैं, सिवाय डिग्री साइन के, क्योंकि वह एक PVALID अक्षर नहीं है। इसलिए, वह इस भाषा के LGR का हिस्सा नहीं बन सकता।

ठीक है, अगले भाग की ओर बढ़ते हैं, जो F अक्षर की आवश्यकता से संबंधित है — यानी F और हुक के साथ F। समुदाय उन्हें समान मानता है, इसलिए यह हमें इन दो अक्षरों के बीच वेरिएंट बनाने का आधार देता है और इस वेरिएंट की डिस्पोज़िशन को ब्लॉक करना चाहिए, इसका कारण यह है कि अगर समुदाय

में एक लेबल एक F अक्षर का उपयोग करके बनाया गया है, तो उसके समकक्ष लेबल, जिसमें दूसरा F इस्तेमाल हुआ हो, सिस्टम में मौजूद नहीं होना चाहिए।

और फिर एक अतिरिक्त लेबल मूल्यांकन नियम की आवश्यकता यह है कि एलोवेरा का क्लिक अक्षर लेबल के शुरुआत में नहीं होना चाहिए। तो, आइए देखें कि यह LGR के रूप में कैसे प्रकट होता है। ठीक है, इस स्लाइड में हम LGR के दो वर्ज़न देख सकते हैं। बाईं ओर है XML वर्ज़न, और दाईं ओर है — XML पूरी तरह से मशीन-पठनीय वर्ज़न है, जबकि HTML अधिकतर मानव-अनुकूल वर्ज़न है, जिसे आप LGR में नेविगेट करने के लिए विभिन्न लिंक का उपयोग कर सकते हैं।

अगर आप XML को ध्यान से देखें, तो आप स्पष्ट रूप से उन चार सेक्शन को देख सकते हैं जिनके बारे में हमने बात की थी। पहला है meta टैग, जिसमें आप LGR से संबंधित सभी मेटा जानकारी देख सकते हैं, इसके बाद है डेटा एलिमेंट, जिसमें कोड पॉइंट की जानकारी होती है और कोड पॉइंट जानकारी के साथ ही वेरिएंट जानकारी भी जुड़ी होती है। तो, आप दो जगह देखेंगे — 0066, जिसमें 0192 का मैपिंग है। और 0192 पर, 0066 की मैपिंग है।

तो, यह फिर से सिमेट्री प्रिंसिपल से आता है — अगर एक अक्षर दूसरे का वेरिएंट है, तो दूसरा भी पहले वाले का वेरिएंट होना चाहिए। और अंतिम खंड में, यानी डेटा तत्व के अंत में, आप नियम सेक्शन देख सकते हैं। हम उस

नियम की एन्कोडिंग देख सकते हैं, जो शब्द के प्रारंभ में है और जो 0192 अक्षर पर लागू होता है। ठीक है, अब अगली स्लाइड पर चलते हैं।

अब जब हमने यह LGR बना लिया है, तो यह कैसा रहेगा। और आइए देखें कि LGR टूल की एक फ़ंक्शन क्या थी — यह लेबल को वैध करने में सक्षम है। तो, अब हम उस चरण में हैं, जहां हम उन विभिन्न लेबलों का सत्यापन कर रहे हैं जिनका उपयोग भाषा समुदाय करना चाह सकता है। तो पहला लेबल है ABE, जिसमें A और B सामान्य अक्षर हैं और E पर एक्यूट है, और यह पूरी तरह से सही निकलता है।

यह एक वैध अक्षर होगा, क्योंकि सभी कोड पॉइंट्स रिपर्टॉयर से हैं और किसी भी प्रकार के नियम का कोई उल्लंघन नहीं है। तो, यह एक वैध अक्षर साबित होगा। दूसरा लेबल है A पर डायरेसिस के साथ, और B, E पहले वाले टेस्ट लेबल जैसे ही हैं। यह अमान्य हो जाएगा, क्योंकि डायरेसिस वाला A इस AG भाषा के कोड पॉइंट रिपर्टॉयर का हिस्सा नहीं है, इसलिए यह इस भाषा के लिए एक अमान्य लेबल होगा। अगली स्लाइड।

फिर वेरिएंट की बात करें तो, आपको याद होगा कि हमने F और F को एक हुक के साथ एक दूसरे के वेरिएंट के रूप में बनाया था। अब अगर हम F A C E लेबल को वैलिडेट करें, जहाँ A पर एक्यूट है, तो इसे वैध लेबल माना जाएगा, लेकिन इसके साथ एक अतिरिक्त जानकारी भी मिलेगी कि इस लेबल का एक वेरिएंट

लेबल भी उपलब्ध है — जिसमें F की जगह हुक वाला F होगा और बाकी अक्षर वही रहेंगे।

और इसका डिस्पोज़िशन ब्लॉक होगा, क्योंकि हमने इसे ब्लॉक्ड वेरिएंट के रूप में मैप किया है। अगर हम दूसरे लेबल की तरफ जाएँ, जिसमें लेबल F हुक के साथ शुरू होता है, तो इसका समकक्ष लेबल, जिसमें F बिना हुक होगा, वह ब्लॉक हो जाएगा। तो, इस बात को थोड़ा और स्पष्ट करने के लिए, यह एक अक्षर को दूसरे पर तरजीह देने का मामला नहीं है। असल में, एक लेबल के भीतर, जो भी अक्षर मौजूद है, वह उपलब्ध रहेगा और उसका दूसरा समकक्ष अक्षर ब्लॉक कर दिया जाएगा। अगली स्लाइड।

अब भाषा के नियम की बात करते हैं। तो चलिए देखते हैं कि यह कैसे काम करता है। तो, आपको याद होगा कि एक अक्षर था, एलोवेरा क्लिक अक्षर, जिसे लेबल के पहले अक्षर के रूप में नहीं होना चाहिए था। तो, CA में और उस विशेष अक्षर के अंत में, यह पूरी तरह से सही निकलता है — यह एक वैध लेबल है, क्योंकि यह किसी भी नियम का उल्लंघन नहीं करता। टेस्ट लेवल 6 उसी विशेष अक्षर से शुरू होता है और इसीलिए इसे अमान्य अक्षर, अमान्य लेबल कहा जाएगा, क्षमा करें।

और फिर इस बात का विवरण दिया जाएगा कि यह विशेष लेबल वैध क्यों नहीं है। आप इसे LGR टूल में देख सकते हैं। अगली स्लाइड। तो, एक समान उदाहरण देने के लिए देवनागरी LGR में — कल के सेशन में भी इसका एक

उदाहरण था। किताब शब्द लिया गया था। इसमें जानबूझकर एक गलती की गई है, जिसमें मात्रा चिह्न दो बार दिया गया है, जबकि इसे केवल एक बार ही दिया जाना चाहिए।

और इसीलिए इस लेबल को अमान्य करार दिया जा रहा है। और इसके पीछे एक सिस्टमैटिक कारण है, जो LGR टूल से निकलता है। फिर वेरिएंट का एक और दिलचस्प मामला है, जिसे संपूर्ण लेबल वेरिएंट कहा जाता है, जिसमें वेरिएंट पूरी स्क्रिप्ट में मौजूद होते हैं। अब, यह एक उदाहरण है जहाँ बाईं ओर का लेबल देवनागरी लेबल है, जो पूरी तरह से देवनागरी अक्षरों से बना है, ठीक उसी तरह जैसा कि पिटिनन ने एपिक उदाहरण में दिया था।

और दाईं ओर गुरमुखी का लेबल है। और एक सामान्य देवनागरी उपयोगकर्ता के लिए, अगर मैं इसे दिखाऊँ, तो वे कह सकते हैं कि यह समान दिखता है, बस कुछ कलात्मक पहलुओं के साथ। इसलिए, वे भी संपूर्ण लेबल वेरिएंट बन जाते हैं। इसी के साथ, मेरा भाग समाप्त होता है। इसे पिटिनन को वापस सौंप दें।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, अक्षत। फिर अगले सेक्शन में, मैं लुइस को आमंत्रित करना चाहूँगा कि वे हमें वर्किंग ग्रुप द्वारा रेफ़रेंस LGR बनाने पर अपडेट दें।

लुइस हूल

आपका बहुत-बहुत धन्यवाद। तो, जो लोग मुझे नहीं जानते, उनको परिचय — मेरा नाम लुइस हूल है, और मैं UCAS वर्किंग ग्रुप का सह-अध्यक्ष हूँ। आज मैं बहुत संक्षेप में बात करूंगा। सबसे पहले, मैं आपसे अनुरोध करना चाहूंगा कि आप अभी इस मैप को देखें। नीले रंग में आप देख सकते हैं, ये Inuit की बोलियाँ हैं, जो कनाडा के उत्तर और पूर्व-उत्तर क्षेत्रों में उपयोग की जाती हैं। संतरी रंग में आप Eastern Cree देखेंगे, और हरे रंग में Western Cree, और गुलाबी और बैंगनी रंग में आपको Dakelh और Dene दिखाई देंगे।

तो, ये भाषाएँ आप अगली स्लाइड में देखेंगे — यह सिलैबिक स्क्रिप्ट वाली हैं। सिलैबिक स्क्रिप्ट में हमारे सिस्टम में 31 भाषाएँ हैं, और लैटिन स्क्रिप्ट में 80 भाषाएँ शामिल हैं। इसका मतलब है 101 और कनाडा में कुल 191 भाषाएँ बोली जाती हैं। तो, यह इसका एक बड़ा हिस्सा है। अब अगले व्यक्ति की बारी है। तो, इस कार्य का दायरा ASCII और फ्रेंच SLD, सेंट-जीन के संदर्भ में परिभाषित करना है।

आप देख सकते हैं एक उदाहरण — francais.quebec, जहाँ एक किला मौजूद है। तो, फिलहाल हमारे पास यही स्थिति है। मूल निवासी लोग डोमेन नाम के लिए अंग्रेजी या फ्रेंच जैसी किसी भाषा का उपयोग कर रहे हैं, लेकिन उनकी अपनी कोई भाषा नहीं है। तो, उदाहरण के लिए, आपके पास naskapi.ca या .quebec या Eastern Cree या कोई भी नाम हो सकता है — जैसे Kariki,

Inuktitut, Inuit या कोई और — उपलब्ध नहीं है। तो, हम इसी पर काम कर रहे हैं।

तो, सिलैबिक और लैटिन वर्किंग ग्रुप के चरण इस प्रकार हैं। इसमें स्थानीय भाषा रिपर्टॉयर का दस्तावेज़ीकरण करना और प्रस्तावित LGR के संपत्ति मान को वैधता की जांच करना शामिल होगा। जो LGR वर्तमान में मौजूद और उपयोग में है, उसे हम कह सकते हैं कि वह पुरानी हो चुकी है, अब अच्छी नहीं है और उपयोग के योग्य नहीं है। फिर, निश्चित रूप से, हमें समकालीन उपयोग की पुष्टि करनी होगी क्योंकि उनकी शब्दावली में कोई भी नया शब्द प्रकट नहीं हुआ है, जिसे इतनी उपयुक्त रूप से कोडित नहीं किया गया हो।

इसलिए, हम लिपिकीय भिन्नताओं, अंतर-लिपि भिन्नताओं, नियमों को परिभाषित कर रहे हैं। हम एक प्रस्ताव को अंतिम रूप देने में लगे हैं, और यहीं मैडम ने XML और HTML निर्माण में बहुत मदद की है। हम सार्वजनिक टिप्पणियों और एकीकरण के लिए तैयार हैं। ये अगले कदम होंगे।

अगली स्लाइड, कृपया। तो, अब सिलैबिक स्क्रिप्ट वाली भाषाओं की ओर बढ़ते हैं। Inuktitut सात उप-भाषाओं, बोलियों या जिसे आप जैसा चाहें, उसके अंतर्गत आती है। इसे समूह द्वारा 2025 में प्रकाशित किया गया है। सिलैबिक स्क्रिप्ट LGR अब पूरा हो चुका है और इसमें अब Inuktitut, Cree और Ojibway भी शामिल हैं। तो यह 18 उपभाषाओं का समूह है। जनता की

प्रतिक्रियाएँ मार्च या अप्रैल 2026 में आने की उम्मीद है। और Dene और Dakelh LGR की उम्मीद इस साल के थोड़े बाद है।

तो, आख़िरी लैटिन स्क्रिप्ट वाली भाषाएँ — अपडेटेड लैटिन स्क्रिप्ट अब पूरी हो चुकी है, जिसमें Nunavut और Labrador Roman Inuktitut शामिल हैं। हम यहां Inuit भाषाओं के बारे में बात कर रहे हैं। और मैं आपको बताना चाहता था कि हम ओंटारियो पहुँच रहे हैं। ओंटारियो का कार्य पूरा हो गया है। तो यह काम कल पूरा हो गया। बचे हुए सिलैबिक स्क्रिप्ट LGR में सिर्फ एक भाषा बाकी है, और लैटिन स्क्रिप्ट में 48 भाषाएँ बाकी हैं। तो, काम अभी खत्म नहीं हुआ है।

अगली स्लाइड, कृपया। मैं उस बॉल को पकड़ता हूँ जो आपने फेंकी थी, और शायद हम जिस पर काम कर रहे हैं उसका एक उदाहरण दूँ। तो, वर्तमान कार्यान्वयन का उद्देश्य निम्नलिखित जोखिमों को कम करना है। तो, इसमें शामिल हैं — होमोग्लिफ, फिशिंग, स्पूफिंग, ट्रेडमार्क, और हम इसमें कुछ और चीज़ें भी जोड़ सकते हैं।

यहाँ हम बात कर रहे हैं टॉप-लेवल सुरक्षा कार्यान्वयन की — ध्यान दें, यह सेकंड-लेवल वर्ड्स के बारे में है, हम किसी भी तरह से फर्स्ट-लेवल एक्सटेंशन की बात नहीं कर रहे हैं। लैटिन स्क्रिप्ट का उदाहरण — आपके पास है le.quebec, LE और फ्रेंच में आपको सभी डायक्रिटिक चिह्नों की संभावनाएँ होती हैं। तो, आप जो भी नाम चुनेंगे, उसमें दो संभावनाएं होंगी। आपका एक

नाम ASCII फॉर्मेट में हो सकता है और दूसरा नाम फ्रेंच भाषा में हो सकता है, जिसमें उच्चारण चिह्न लगा हो। यह कोई भी एक्सेंट हो सकता है या सभी हो सकते हैं, इसलिए जो अन्य डायक्रिटिक चिह्न बचते हैं, उन्हें ब्लॉक करना होगा।

इनका उपयोग नहीं किया जा सकता है और इन्हें आपस में जोड़ा भी नहीं जा सकता है। इसलिए, आपको ऐसा टूल चाहिए जो इसे सही ढंग से कर सके। इसका परिणाम यह होगा कि geoTLD दायरे में दुरुपयोग की संभावना कम हो जाएगी, क्योंकि हमें यह मानना होगा कि कुछ शब्दों में ऐसा डायक्रिटिक हो सकता है जिसे अनुमति दी जा सकती है, लेकिन आप उसे अनुमति नहीं देना चाहते, क्योंकि इससे सिस्टम में भ्रम पैदा हो सकता है। इसलिए, निश्चित रूप से, जनहित में सुरक्षित DNS ही है। अगला वाला।

हम अब उसी चरण में हैं, अपने कार्य के अंत की ओर, और हमें आशा है कि इसका परिणाम एक अधिक खुला इंटरनेट होगा। तो, अगले कुछ हफ्तों में हमारी एक मीटिंग होने वाली है, और हमें आशा है कि इस कार्य का परिणाम अधिक लोगों के लिए इंटरनेट, यानी कनाडा में Inuit और First Nation के लिए इंटरनेट होगा। मुझे लगता है कि अब उनके पास कोई असटीक मोड नहीं है। और जैसा कि आप जानते हैं, इंटरनेट सभी के लिए है। और जैसा कि कहा जाता है, एक दुनिया और एक इंटरनेट। धन्यवाद।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, लुइस। हमें अपडेट देने के लिए धन्यवाद। फिर अगले में, मैं इंडोनेशिया से घेअफ़नी को आमंत्रित करना चाहूँगी, जो रिमोटली बोलेंगी और JFR इंडोनेशियाई स्क्रिप्ट के रेफ़रेंस पर अपडेट देंगी। अब आपकी बारी है, घेअ।

घेअफ़नी विद्यातिका पुत्री

बहुत धन्यवाद, पिटिनन। नमस्ते, सभी को। मैं घेअ हूँ, PENDI से, वह संगठन जो indonesians.id डोमेन का प्रबंधन करता है। तो, आज मैं इंडोनेशियाई स्क्रिप्ट्स पर एक त्वरित समुदाय अपडेट साझा करूंगी, विशेष रूप से बालीनीज़, जावानीज़ और पगन, और यह कि हम इन स्क्रिप्ट्स को डोमेन नामों में उपयोग योग्य बनाने में कैसे मदद कर रहे हैं। अगली स्लाइड, कृपया। चलिए एक सरल विचार से शुरुआत करते हैं।

डोमेन नाम आपका डिजिटल पता होता है। अगर आपका डिजिटल पता केवल एक ही लिपि में लिखा जा सकता है, तो कई लोगों को लगता है कि इंटरनेट उनकी भाषा नहीं समझता है। इंडोनेशिया में जीवित स्थानीय स्क्रिप्ट्स हैं, ये संग्रहालय की वस्तुएँ नहीं हैं — इन्हें स्कूलों, सांस्कृतिक गतिविधियों और समुदायों में इस्तेमाल किया जाता है। तो, IDN महत्वपूर्ण हैं, क्योंकि ये लोगों को अपने अकाउंट्स ऑनलाइन इस्तेमाल करने की अनुमति देते हैं।

लेकिन यहाँ मुख्य बात यह है — अगर हम इसे अनुमति दें और देखें, इसे ऐसे सोचें जैसे साइन की तरह, अगर दो संकेत लगभग समान दिखते हैं, तो लोग

भटक सकते हैं या उससे भी बुरा, धोखा खा सकते हैं। इसलिए, हमारा मिशन उपयोगकर्ताओं की सुरक्षा सुनिश्चित करते हुए उन्हें प्रेरित करने में सक्षम बनाना है। यही वह जगह है जहाँ रेफ़रेंस LGR का काम आता है।

अगली स्लाइड, कृपया। सारांश में, एक रेफ़रेंस LGR किसी स्क्रिप्ट के लिए तीन व्यावहारिक प्रश्नों का उत्तर देता है, जैसा कि पिटिनन पहले ही समझा चुकी हैं। पहला सवाल यह है कि किन अक्षरों का उपयोग किया जा सकता है और फिर किन क्रमों का पालन किया जाता है, क्या सही दिखता है और व्यवहार में काम करता है, और फिर किन मामलों में भ्रम हो सकता है। इसलिए, उन्हें विभिन्न प्रकार के सुरक्षा उपायों की आवश्यकता है। महत्वपूर्ण बात यह है कि यह काम केवल एक संगठन द्वारा नहीं किया जाता है, बल्कि यह समुदाय द्वारा संचालित होता है। तो, हमारा तरीका यह है कि हम FGD (फोकस ग्रुप डिस्कशन) आयोजित करते हैं, जिसमें समुदाय के हितधारक जैसे विशेषज्ञ, प्रैक्टिशनर, अकादमिक और समुदाय के प्रतिनिधि शामिल होते हैं, और हम वास्तविक लेखन प्रथाएँ दर्ज करते हैं — कि लोग वास्तव में क्या लिखेंगे, क्या स्वीकार्य है और क्या नहीं।

और फिर हमारे पास कुछ ऐसा है जो प्रक्रिया को आगे बढ़ाने में वाकई मदद करता है। हम इसे सामुदायिक चैंपियन कहते हैं। यह एक विश्वसनीय केंद्र व्यक्ति होता है और उसके बाद एक सम्मानित व्यवसायी या शिक्षाविद होता है जो दोहरी भूमिका निभाता है। पहला व्यक्ति मध्यस्थ की भूमिका निभाता

है, और जब अलग-अलग दृष्टिकोण सामने आते हैं, तो दूसरा व्यक्ति मुख्य संपादक होता है जो दस्तावेज़ को विभिन्न संस्करणों में सुसंगत बनाए रखता है, परिणामों का समन्वय और दस्तावेज़ीकरण करता है और ICANN के साथ संपर्क स्थापित करता है, तथा ICANN, LGR ढांचे के साथ संरेखण की समीक्षा करता है और प्रकाशन चरणों का मार्गदर्शन करता है।

तो, कार्यप्रणाली में पहले सामुदायिक अभ्यास, फिर दस्तावेज़ीकृत नियम, एक पुनरावृत्ति समीक्षा और आखिर में प्रकाशन प्रकार शामिल हैं। और असल में, चैंपियन के बिना, आपको मेरी राय के 30 संस्करण मिलेंगे। चैंपियन के साथ, आपको हमारी आम सहमति का एक संस्करण मिलता है।

अगली स्लाइड, कृपया। अब, आइए स्क्रिप्ट के अनुसार तेजी से आगे बढ़ें। पहली है बाली भाषा की स्क्रिप्ट। मैं कह सकता हूँ कि बाली भाषा इंडोनेशिया के लिए एक महत्वपूर्ण उपलब्धि है क्योंकि यह पहली इंडोनेशियाई स्थानीय लिपि है जिसने नवंबर 2024 में ICANN द्वारा प्रकाशित द्वितीय-स्तरीय संदर्भ LGR का मील का पत्थर हासिल किया है। और SLD को 11 दिसंबर 2025 को बाली के राज्यपाल के हाथों लॉन्च किया गया था।

और दूसरी है जावानीज़। जावानीज़ भाषा के लिए भी हमारे पास मजबूत गति है। ICANN ने पिछले साल, विशेष रूप से पिटिनन ने योग्याकार्ता का दौरा किया था, ताकि स्थानीय हितधारकों के साथ वास्तविक उपयोग का प्रत्यक्ष अवलोकन किया जा सके, और जावानीज़ संदर्भ LGR पहले ही प्रस्तुत किया

जा चुका है। प्रकाशन की प्रक्रिया के तहत इसे वर्तमान में ICANN के साथ अंतिम रूप दिया जा रहा है, जिसके बाद सार्वजनिक टिप्पणी का चरण शुरू होगा। मुझे लगता है कि यह इस बात का एक अच्छा उदाहरण है कि सामुदायिक सहभागिता क्यों महत्वपूर्ण है। हम नियमों को अलग-थलग करके नहीं बना रहे हैं, बल्कि हम उन्हें इस बात के अनुरूप बना रहे हैं कि असल में स्क्रिप्ट का उपयोग कैसे किया जाता है।

और आखिरी वाली है पेगॉन स्क्रिप्ट। पेगॉन के लिए, संदर्भ बीजगणित सबमिट कर दिया गया है और हम ICANN के फ़ीडबैक का इंतज़ार कर रहे हैं। पेगॉन का उपयोग विभिन्न समुदायों में सक्रिय रूप से किया जाता है, इसलिए अगला कदम समीक्षा प्रतिक्रियाओं को शामिल करना और नियमों को अधिक व्यावहारिक और सुरक्षित बनाए रखने के लिए हितधारकों के साथ मिलकर काम करना है। अगली स्लाइड, कृपया।

यहां एक सरल उदाहरण के माध्यम से दिखाया गया है कि हम सुरक्षा और बीमा योग्यता के बीच संतुलन कैसे बनाते हैं। ठीक है, प्रमुख पात्रों को अक्सर जल्दी पढ़ा जा सकता है। संदेश को या तो छोटी स्क्रीन पर या मुद्रित सामग्री पर देखें, इसलिए दृश्य भ्रम एक वास्तविक समस्या है। जावानीज़ स्क्रिप्ट में, कुछ अंक अक्षरों से मिलते-जुलते हो सकते हैं। इसलिए, हमारे कार्य समूह ने एक रूढ़िवादी दृष्टिकोण अपनाया। हम अंकों को शामिल नहीं करते हैं, लेकिन भ्रम और संभावित दुरुपयोग को कम करने के लिए ASCII अंक और जावानीज़

अंक शामिल करते हैं। एक उदाहरण के तौर पर, जैसा कि आप देख सकते हैं, जावानीज़ अंक 1 जावानीज़ अक्षर 'गो' के समान दिख सकता है। यदि उपयोगकर्ता आसानी से अंतर बता सकते हैं, तो इससे इस प्रकार की गलतियाँ, गलत डिलीवरी या प्रतिरूपण जैसी स्थितियां पैदा होती हैं।

अब, क्या इससे उपयोगिता कम हो जाती है? शायद, हाँ। इसलिए हम एक उपयोग में आसान बैकग्राउंड उपलब्ध कराते हैं। लेबल में संख्याओं को शब्दों के रूप में लिखें। तो, अगर आप ICANN1 जैसा डोमेन बनाना चाहते हैं, तो आप इसे ICANONE लिख सकते हैं, जैसे कि इस तरह। और इसी तरह हम संतुलन बनाए रखते हैं। हमने उपयोगकर्ताओं के लिए आधारभूत जानकारी सहेजने के तरीके से शुरुआत की, और फिर वास्तविक लेखन प्रथाओं के साथ तालमेल बिठाकर लेबल को व्यावहारिक बनाए रखा, और समुदाय की प्रतिक्रिया और वास्तविक उपयोग से सीखते हुए समय के साथ इसमें संशोधन किया। अगली स्लाइड, कृपया।

आखिर में, आगे क्या होगा? जावानीज़ भाषा के लिए, हम टोआ प्रकाशन और सार्वजनिक टिप्पणियों के साथ कार्यान्वयन को पूरा करते हैं। पेगॉन के लिए, हम ICANN की प्रतिक्रिया के आधार पर इसमें सुधार कर सकते हैं। और बाली में, हम व्यावहारिक मार्गदर्शन और संपर्क के माध्यम से इस प्रक्रिया को और मज़बूत करना जारी रखते हैं। और कुल मिलाकर, दरअसल हम एक सतत विकास प्रक्रिया चाहते हैं ताकि जब समुदाय तैयार हो जाए तो बाटक, सुंडानी

या बुगिनीज़ जैसी दूसरी इंडोनेशियाई स्क्रिप्ट भी इसमें शामिल हो सकें। मेरे ख्याल से बस इतना ही है। धन्यवाद, पिटिनन। वापस आपके पास।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, घेअ। और मुझे लगता है कि इसी के साथ हमारे प्रस्तुतकर्ताओं का सत्र समाप्त होता है। तो अब, मैं आप सभी के सवालों या टिप्पणियों के लिए मंच खोलना चाहूंगी। अगर आप बोल रहे हैं, तो कृपया रिकॉर्ड के लिए अपना नाम और संबद्धता बताएं। धन्यवाद।

दीपक परमार

रिकॉर्ड के लिए, मेरा नाम है दीपक परमार है। मेरा एक बुनियादी सवाल है कि अंतिम उपयोगकर्ता डोमेन नाम टाइप कर रहे हैं या आवाज से, क्योंकि इनपुट स्तर में बहुत बड़ा अंतर होता है, अगर मैं कुछ बोल रहा हूँ, तो मान लीजिए मेरी अंग्रेजी भारतीय लहजे वाली है और दूसरे देश की अंग्रेजी यूरोपीय लहजे वाली है, तो इसी तरह आगे भी, हम कोड के दो तरीकों के बारे में जो बहस कर रहे हैं, वह इनपुट के रूप में कैसे आएगा, क्योंकि हमारी मातृभाषा हमेशा किसी न किसी भाषा में निहित होती है। इसलिए, अगर यह किसी डोमेन के लिए आवाज़-आधारित इनपुट है, तो हमें असल में सारी चीज़ों पर फिर से सोचने की ज़रूरत होगी।

लुइस हूल

ICANN प्रथम राष्ट्र और Inuit भाषाओं के साथ अपने अनुभव के आधार पर थोड़ी बात करता है। हमारे कार्य समूह में कीबोर्ड के विशेषज्ञ लोग हैं, और वे कीबोर्ड से संबंधित उन सभी समस्याओं को दूर करने में हमारी मदद कर रहे हैं जो वर्तमान समय में भी मौजूद हो सकती हैं या कुछ ऐसे वर्णों के उपयोग से संबंधित हो सकती हैं जो अप्रचलित या अप्रचलित हैं, अब उपयोग में नहीं हैं। तो, मैं सिर्फ़ एक व्यक्ति की बात नहीं कर रहा हूँ, बल्कि हम दुनिया में कीबोर्ड के क्षेत्र में अग्रणी लोगों की बात कर रहे हैं, इसलिए वे केवल UCAS कीबोर्ड ही नहीं बनाते हैं।

दीपक परमार

इस बारे में थोड़ी और जानकारी। हमारे विशेषज्ञ तो विशेषज्ञ हैं, लेकिन अंतिम उपयोगकर्ता विशेषज्ञ नहीं होते।

पिटिनन कूआर्मोर्नपटाना

हो सकता है कि मैं आपके सवाल में थोड़ा सा और जोड़ दूं, कि अगर आवाज़ को इनपुट के रूप में उपयोग किया जाए तो यह कैसा होगा? तो, यह प्रोजेक्ट मूल रूप से डोमेन नामों से संबंधित है और फ़िलहाल इसे विकसित करने की प्रक्रिया विज़ुअल आधार पर आगे बढ़ रही है। इसके बाद, वास्तव में जो इनपुट मिलता है, चाहे वह कीबोर्ड से हो या जब आप बोलते हैं, वही आखिरकार टेक्स्ट में भी बदल जाता है।

तो, फिलहाल, यह नियम विज़ुअल पर आधारित है और समुदाय के कुछ हिस्सों में, शायद जब अर्थ के हिसाब से भिन्नता क्या है, इस पर स्पष्ट नियम बन जाएंगे, तो इसे उस तरह से भी परिभाषित किया जा सकता है। लेकिन जाहिर है, फ़िलहाल हम आवाज़ या लहज़े का दायरा नहीं बढ़ाएंगे, लेकिन आपके सुझावों के लिए धन्यवाद। यह इनपुट स्तर पर एक ऐसी चीज़ है जो शायद LGR के दायरे में नहीं आती। धन्यवाद।

मोहम्मद ज़ुल्कर नायन

नमस्ते, मेरा नाम ज़ुल्कर है। मैं ISOC ओंटारियो से हूँ। मेरा लुइस से एक छोटा सा सवाल है। तो, इंटरनेट के लिए स्वदेशी कनाडाई पाठ्यक्रम पर आप जो काम कर रहे हैं, उसके बारे में मुझे बस यह जानना है कि क्या स्वदेशी समुदाय ही इसकी मांग को बढ़ावा दे रहा है? या फिर यह कुछ ऐसा है जिसे आप बना रहे हैं, आपकी टीम अभी बना रही है, और वे बाद में इसमें शामिल होंगे? स्थिति क्या है?

लुइस हूल

दोनों। यह भाषा पर निर्भर करता है। यह क्षेत्र पर निर्भर करता है। क्योंकि हमारा इरादा बहुत सरल और सीमित था। हम सिर्फ .quebec रजिस्ट्री के ज़रिए क्यूबेक में बोली जाने वाली भाषाओं को उपलब्ध कराना चाहते थे। तो, हमने Inuit से शुरुआत की। इन सात बोलियों में से केवल दो ही क्यूबेक में बोली जाती हैं।

अगर मैं दूसरी भाषाओं को लूं, तो फिर से, आपके पास क्री है, [00:53:21 – अस्पष्ट], बस। लेकिन समस्या यह है कि कनाडा में, जैसा कि आप जानते हैं, फर्स्ट नेशन, जैसा कि आपने मेरी पहली स्लाइड में देखा, आपको वृत्त दिखाई देते हैं, और यह क्यूबेक है। Inuit लोगों के लिए, यह दायरा क्यूबेक से कहीं अधिक व्यापक है।

और अगर आप उन भाषाओं में से किसी को भी लें जिनपर हम विचार कर रहे हैं, जैसे कि क्यूबेक, ओंटारियो, मैरीटाइम्स, ब्रिटिश कोलंबिया, आदि, तो आप उस योजना पर उस तरह से काम नहीं कर सकते क्योंकि यह लागू नहीं होगी। तो, असल में, हमने एक ऐसी बात का पता लगाया जो कई लोगों के लिए स्पष्ट थी, लेकिन हम भोले थे। अगर आप इसे शुरू करते हैं, तो आप इसे खत्म करना चाहेंगे।

मोहम्मद ज़ुल्कर नायन धन्यवाद।

पिटिनन कूआर्मोर्नपटाना हाँ, कृपया अपनी बात कहें।

लांस हिंड्स धन्यवाद, अध्यक्ष महोदय। लांस हिंड्स, LACRALO के अध्यक्ष, और मेरी दूसरी भूमिका में, एक IT NGO, असल में यह देखने की कोशिश कर रहे हैं कि

हम कैरेबियन स्थित गुयाना में भाषा संरक्षण और सांस्कृतिक संरक्षण को आगे बढ़ाने के लिए तकनीकों का कितना उपयोग कर सकते हैं। फर्स्ट नेशन के साथ परामर्श के संबंध में, मेरी उत्सुकता यह जानने में है कि ज़मीनी स्तर पर क्या होता है, क्योंकि, उदाहरण के लिए, हमारे पास वह कौशल नहीं है।

इसलिए, दस्तावेज़ीकरण की कमी के कारण हमें कभी-कभी भाषा में मौजूद चिह्नों की पहचान करने के लिए वास्तव में गहन अध्ययन करना पड़ता है। इसलिए, आपको टकराव का सामना करना पड़ता है। तो, उदाहरण के लिए, आपके पास एक दाएं एकल उद्धरण चिह्न जैसा कुछ होगा, न कि कोई संशोधक अक्षर, एपोस्ट्रोफ़ी या वास्तव में जो एक स्वरयंत्र विराम चिह्न है। और मैं आपके अनुभव के बारे में जानने को उत्सुक हूँ, इस संबंध में -- तो, नियम कहता है कि आपको समुदाय के पास वापस जाना चाहिए, और वे ही तय करेंगे कि कौन सा सबसे अच्छा है।

अब, आखिर में होता यह है कि अगर आप कहते हैं, ठीक है, ICANN का भाषाई एपोस्ट्रोफ़ी के लिए पसंदीदा रूप, तो कोई कहता है, ठीक है, सबसे अच्छे तरीके का पालन करें और उसका उपयोग करें। लेकिन फिर आखिर में कोई आपत्ति जता सकता है और कह सकता है, ठीक है, आप यह अकेले नहीं कर सकते। तो, मुझे बस यह जानना है कि आपके अनुभव के आधार पर, जब आप टकराव की स्थिति में फ़ंसते हैं, तो आप उससे कैसे निपटते हैं।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद। और यह एक बहुत ही जायज सवाल है। और उम्मीद है कि हमारे पास इसका एक अच्छा उदाहरण होगा। तो, अगर हम इन सभी चीजों को सक्षम करने के मूल सिद्धांतों पर वापस जाएं, तो पहला कदम यह सुनिश्चित करना है कि ये सभी लेबल IDNA2008 के अनुरूप हों।

तो अगर हम उस पर वापस जाएं, तो इनमें से कई एकल उद्धरण चिह्न या कुछ भाषाओं में विस्मयादिबोधक चिह्न, जिसका उपयोग क्लिक ध्वनि या नासिका ध्वनि के रूप में किया जाता है, वास्तव में अस्वीकृत गुण में है, क्योंकि इसका उपयोग प्रोटोकॉल में अन्य वाक्य संरचना में भी किया जाता है। तो फिलहाल स्थिति यही है। इस किरदार का कुछ हिस्सा वास्तव में PVALID की संपत्ति नहीं है। लेकिन मुझे लगता है कि लुइस भी इस बारे में बता सकता है कि उसने क्या पाया और वह आगे क्या करने वाला है।

लुइस हूल

हाँ, और हमने उस अभ्यास के दौरान कई बार PVALID के साथ प्रयोग किया है। मुझे बस इतना और कहना है कि जहां तक मेरा सवाल है, मैं इनमें से किसी भी भाषा का विशेषज्ञ नहीं हूँ। मेरी विशेषज्ञता फ़्रेंच भाषा में है। लेकिन आपको बता दें, उस समूह में हमारे पास ऐसे लोग हैं जो उन मुद्दों का जवाब देने में सक्षम हैं। हमारे समूह में लोग हैं।

उदाहरण के लिए, बिल ने 2025 में अनास्तासिया पी. डिक्शनरी प्रकाशित की। इसलिए, हम यह मान लेते हैं कि वे जो कह रहे हैं, उसके बारे में उन्हें जानकारी

है और हम PVALID के मान्य होने या न होने से संबंधित किसी भी मुद्दे या सवाल के लिए समुदाय के साथ व्यवस्थित रूप से संपर्क में रहे हैं।

इसके उपयोग से संबंधित सभी सवाल, क्या ये सब स्कूल में पढ़ाया जाता है? क्या यह आमतौर पर उपयोग में है? और A या B में से किसी एक को चुनने का फ़ैसला हमने नहीं किया। दरअसल, यह फैसला समुदाय ने किया। दिन के आखिर में, बस यही हुआ था।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, लांस। सवाल के लिए आपका धन्यवाद। और हमने एक हाथ उठाया है। बिल, क्या आप अपना माइक अनम्यूट करना चाहेंगे?

बिल जौरिस

रिकॉर्ड के लिए, मेरा नाम बिल जौरिस है। मुझे लगता है कि इस समस्या का एक हिस्सा इस तथ्य से उत्पन्न होता है कि IDN प्रोजेक्ट में शामिल लोग, और उससे भी अधिक, IETF में शामिल लोग जिन्होंने इन सभी चीजों के लिए मानक लिखे, उनमें से कोई भी ऐसी भाषा का मूल वक्ता नहीं था जिसमें क्लिक या ग्लोटल स्टॉप शामिल हों और इसलिए, उन्होंने इन सभी को नजरअंदाज़ करने का निर्णय लेने में काफ़ी लापरवाही बरती।

नाममात्र का कारण यह है कि वे विराम चिह्नों की तरह दिखते हैं, लेकिन वास्तविकता यह है कि इससे बचा जा सकता था, सिवाय इसके कि नियम

बनाने वालों ने इसे महत्वपूर्ण नहीं समझा। इसलिए, उदाहरण के लिए, ग्लोटल स्टॉप के रूप में एपोस्ट्रोफ़ी का उपयोग वर्जित है। नियम बनाने वालों को इस बात की बिल्कुल परवाह नहीं थी। ऐसा कहना शायद उनके साथ अन्याय होगा, लेकिन मेरी नजर में तो स्थिति यही है। धन्यवाद।

पिटिनन कूआर्मोर्नपटाना

धन्यवाद, बिल और आपकी टिप्पणी के लिए भी धन्यवाद। तो, इसमें यह जोड़ना चाहूंगी कि असल में और इनके अलावा भी कई चीज़ें हैं जिनकी अनुमति नहीं है। उदाहरण के लिए, कुछ स्क्रिप्ट में, रोज़मर्रा की भाषा में दो अक्षरों को आपस में जोड़ने के लिए उन्हें ज़ीरो विड्थ नॉन-जॉइनर की आवश्यकता पड़ सकती है। लेकिन क्योंकि यह अदृश्य है, इसलिए इसे शीर्ष-स्तरीय डोमेन में अनुमति नहीं है, उदाहरण के लिए।

क्योंकि यह भी एक साझा संसाधन है, इसलिए हमने डिजाइन में IETF द्वारा विकसित मानक को आधार बनाया है। और मुझे लगता है कि इसका मकसद इसे सुरक्षित रखना है। जाहिर है, अगर इस या वैकल्पिक समाधान का उपयोग करने की मांग होती है, तो हम इस बारे में बातचीत शुरू करना चाहेंगे और देखेंगे कि हम मिलकर कोई समाधान कैसे निकाल सकते हैं। लेकिन आपके सुझाव के लिए धन्यवाद, बिल। आपके सुझावों के लिए धन्यवाद।

चंपिका विजयतुंगा	शायद मुझे लगता है कि हम उस समय तक पहुंच गए हैं, और सवाल-जवाब पॉड में भी कोई सवाल नहीं हैं। तो, मुझे लगता है कि हम सेशन को खत्म कर सकते हैं। इस सेशन में सभी वक्ताओं और प्रतिभागियों को बहुत-बहुत धन्यवाद। धन्यवाद, और अब हम रिकॉर्डिंग बंद कर सकते हैं। धन्यवाद।
पिटिनन कूआर्मोर्नपटाना	आप सभी को धन्यवाद। सभी वक्ताओं को धन्यवाद। हमारे बीच आने के लिए धन्यवाद।

[ट्रांसक्रिप्शन यहां खत्म होता है]