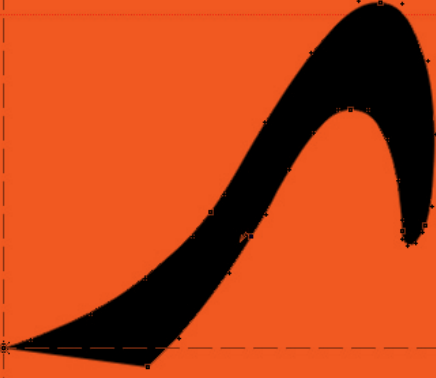


ستيفان بوف | ترجمة: يارا خوري

# ملف تعليمي حول انتاج الفونت العربي

الجزء الثاني: الفونت المخطوط



هذا الملف التعليمي هو حصيلة مقابلة قام بها ايدو سميتسهوزن مع ستيفان بوف. وبينما كان المصممان يعملان معاً على المقابلة، نشأت فكرة انتاج ملف تعليمي طال انتظاره حول كيفية انتاج فونت (خط رقمي) عربي. قام ستيفان بوف بكتابة نص هذا الملف بينما كان ايدو سميتسهوزن مصوتا له.

عمل ستيفان كمهندس كمبيوتر لاكثر من ١٥ سنة. درس اللغة العربية في جامعة غرينوبل وواصل الدراسة في المركز الوطني للتعليم عن بعد (CNED). التحق بـ«وينسوفت انترناشونال» (Winsoft International) عام ٢٠٠٦. ومنذ ذلك الوقت عمل على نسخ الشرق الأوسط للـ«فايل مايكر» (FileMaker) و«الادوبي دريم ويفر» (Adobe Dream Weaver) والى حد ما على الـ«إنديزاين» (InDesign). كان مختصاً ايضاً بتصميم المحرك الذي يعتبر أساس كل منتجات الشرق الأوسط. وبالأخص، اضاف ستيفان بوف الى المحرك الـ«أوبن تايب» (OpenType) دعماً للنصوص الهندية ونصوص جنوب شرق آسيا.

يعتبر هذا الملف الثاني من ثلاثة. ويتناول الملف الأول الخطوط الطباعية والثالث خطوط الإنترنت الطباعية.

حقوق التأليف © ٢٠١١ ستيفان بوف

الطبعة الأولى ٢٠١١  
حقوق النشر © Khatt Books (دار خط للكتب)

Khatt Books  
Van Tuyll Van Serooskerkenweg  
١٠٧٦ JT Amsterdam  
The Netherlands  
www.khattbooks.com

ترجمة النص: يارا خوري  
تصميم الكتاب: هدى سميتسهوزن أبي فارس.

الفونت المستخدم في هذا الكتاب: «فدرا عربي» من تصميم بيتر بيلاك (صمم في البدء بالتعاون مع طارق عتريسي)، لمشروع «المراوحة التيبوغرافية» لـ«مؤسسة خط»، عام ٢٠٠٧.

جميع الحقوق محفوظة.  
يمنع استخدام أو نسخ محتويات هذا الكتاب بأي شكل أو وسيلة من غير موافقة الناشر الخطية.

المعلومات الموجودة في هذا الكتاب موزعة من غير ضمانات. مع أنه قد تم الأخذ بجميع الاعتبارات من أجل ضمان دقة وصحة المعلومات المحتواة في هذا الكتاب، فإن المؤلف والناشر غير مسؤولين تجاه أي شخص أو جهة لتجاهل حقوقهم للتأليف والنشر عن غير قصد. و يسرنا ذكر أي إثمات ضرورية في الطباعات اللاحقة.

## مقدمة

تناولنا في الملف التعليمي الأول أسس عمل «أوبن تايب» (OpenType). أما هذا الملف فيخاطو خطوة إضافية إلى الأمام، ويشرح ميزات متقدمة من «أوبن تايب» واستراتيجيتها. وسنناقش على الخصوص مفاهيم المضمون (context)، والمتغيّرات (alternates)، والمعالجة أخرى لتموضع النقط. ونفترض في هذا الملف أنكم ملّمون بمفهوم «فولت» (VOLT)، وقادرون على خلق ميزات (features) ولوائح البحث (lookups) الخ.

## تصميم فونت عربي مخطوط

إن أنواع الخطوط الرقمية أو الفونت التي يمكننا تصميمها باستخدام الميزات التي وصفناها في الجزء الأول من الملف التعليمي، محدودة جداً. فمثلاً، لا يمكن تمثيل الشكل الموضوعي الواحد إلا برمز (glyph) واحد. لكن حين نحاول أن نصمم فوتتات أكثر مخطوطية (calligraphic) بطبيعتها، نحتاج إلى أكثر من رمز لشكل موضوعي واحد. مثلاً، لنأخذ كلمة «بحر» نموذجاً. نلاحظ هاهنا أن الشكل المطبوع بسيط نوعاً ما، ويفضّل القارئ المتعود على أنماط الكتابة العربية التقليدية، ان يرى شيئاً أشبه بما يلي:

بحر بحر بحر

خطوط عربية مختلفة، من اليسار: الثلث والرقعي والنسخي.

إن غرض هذا الملف التعليمي هو شرح التقنيات المطلوبة التي تمكننا من الوصول الى نتائج مخطوطة كهذه. لتتمكن من ذلك، نحتاج إلى الطلاع على المفاهيم الجديدة التالية:

1. ميزة التفكيك (decomposition feature)
2. ميزة المتغيّرات (alternate feature)
3. ميزة المضمون (context feature)
4. ميزة <calt>
5. ميزة تموضع الحروف المتصلة (cursive positioning featurure)



كما ترى، لا تحتاج هنا إلا إلى أن نستبدل الرمز الذي يمثل الشكل المنفصل لكل حرف، بجزئين منفصلين: متن الحرف (ويدعى أحياناً grapheme)، و النقطة أو حركة التشكيل. تذكر في نهاية هذا العمل أنك تحتاج إلى وضع النقطة في مكانها، وتأكد إذن من فصل النقط التي تأتي فوق المتن عن تلك التي تحت المتن، حتى لو اضطرت إلى تكرار الرمز ذاته. فإذا لم تفعل، لن تتمكن من التمييز بين حرفي التاء والياء الأوليين («ت» و«ي»)، لأن كليهما شكل الرمز نفسه. ولوضع النقاط في مواقعها الصحيحة (فوق متن الحرف أو تحته)، لا بد من تسمية كل منهما تسمية مختلفة، مع أن شكل رمزيهما واحد. لمتابعة فكرة انتاج الفونتات المخطوطة هذه، عليك أن ترسم أشكال الحروف مجردة من نقاطها، وأن تخلق مجموعة منفصلة من رموز النقاط المعزولة. أخيراً عليك أن تخلق رموزاً «كاملة» (لها قيمة الـ Unicode) للرموز المنفصلة، تعمل كنقطة الابتداء. ولكن حالما تطبق ميزة <ccmp>، تُستبدل الرموز الأصلية، ولا تعود تُستخدم.

ملاحظة: يُستحسن خلق نسخة ثانية أيضاً للرموز التي تخلو من النقاط (مثل رمز الألف «أ»)، وفي نموذجنا تُستبدل الألف (التي تحمل الشيفرة U0627)، ليحل محلها رمز مطابق يدعى (g2071\_s452). إن الهدف من هذا هو الفصل التام بين الرموز المستخدمة قبل تطبيق ميزة <ccmp>، وتلك المستخدمة بعد تطبيقها.

## ٢. ميزة المتغيرات

الآن وقد نجينا النقاط، يمكننا التركيز تماماً على أشكال الرموز (glyph shapes). في هذا الملف التعليمي، نحصر الأشكال المتغيرة بشكلين لكل حرف؛ الشكل «المألوف» (أو المقياسي)، وشكل آخر مخصص ليستخدم قبل حرف الجيم «ج». أما إذا شئت أن تصمم فونتاً مخطوطاً كاملاً، فستحتاج حتماً إلى عشرات الأشكال المتغيرة لكل رمز الحرف الواحد.

# تأخ تحت ترس تص تفقى

في التعبيرات المستخدمة في OpenType، تسمى هذه الأشكال المتغيرات (alternates)؛ ويحتوي OpenType نوعين:

١. متغيرات الأسلوب (stylistic alternates)، حيث يمكن للمستخدم انتقاء شكل الرمز اختياريًا لأسباب جمالية.
٢. متغيرات المضمون (contextual alternates)، حيث يكون انتقاء الشكل المتغير آلياً، بحكم المضمون (مثل الرموز التي تظهر قبل و/أو بعد رمز ما).

إن استخدام متغيرات الأسلوب <salt> سهل. والشرط الوحيد في تصميمها هي أنها تتطلب جميعاً الوصلة ذاتها بين الحروف. وحتى نستخدم متغيرات الأسلوب (stylistic alternates)، نحتاج إلى اختيار لائحة المتغيرات، ثم ننتقي الرمز المطلوب من الرموز المتاحة في اللائحة. ولا تطبق محركات النص هذه الميزة آلياً. وحدها محركات النص المتطورة جداً (مثل محرك نص «إنديزاين» نسخة شرق الأوسط (InDesign ME)) تتيح لك بطريقة محددة أن تطبق الميزة على حرف واحد. لذا فإن القدرة على استخدام ميزة <salt>، تبقى اختيارية.

في هذا الملف التعليمي، نركز فقط على متغيرات المضمون (contextual alternates)، ونترك متغيرات الأسلوب جانباً. وتتطلب متغيرات المضمون تعريف «المضمون»، وطالما أن متغيرات كثيرة قد تحدث، فإن تراتبية تطبيق هذه المتغيرات دقيقاً ومعقداً أحياناً. فلنلق إذن نظرة عن كثب على ميزة المضمون.

### ٣. ميزة المضمون

المضمون، كما يعرفه OpenType، هو تحديد أي رمز (أو رموز) يمكن أن تسبق أو تلي رمزاً ما. أما في (VOLT)، لتحديد المضمون ثمة منطقة معينة لكل لائحة بحث (lookup):

Context Before	Context After
>	<96_12_in>

الخط العمودي الذي يظهر في المضمون يعيّن الموضع الحالي، وبما أننا نعمل بلوائح بحث، حيث يتجه النص فيها من اليمين إلى اليسار (Right-To-Left lookups)، فإن ما يظهر إلى الجانب الأيسر يحدد ما يلي، وما يظهر إلى الجانب الأيمن يحدد ما يسبق. في النموذج المبين أعلاه، يمكن أن تترجم ميزة المضمون إلى: «نحن هنا أمام شكل حاء (ح) وسطية».

زاوية المضمون في اللائحة (lookup).

وبما أن هذا الفونت يحتوي على عدة أشكال للحاء الوسطية، فقد استخدمت مجموعة لها الشكل نفسه. ويمكنك أن تعيّن المضمون باستخدام رمز (glyph) واحد، أو مجموعة رموز، أو رموز كثيرة أو مجموعات كثيرة. ولحصر عدد ال lookups، تحتاج إلى تعيين مضامينك بعناية، لتجعلها شاملة قدر المستطاع.

ح	g2672_5497
ا	g2744_5929

يمكنك أيضاً أن تحدد المضامين التي تشتمل على الرمز الحالي، مثلاً، الأسلوب التقليدي لكتابة كلمة «بيت» هو باستخدام ياء «ب» وسطية مطوّلة مثل هذه: ولأجل ذلك، عليك أن تخلق المضمون الآتي:

بداية مضمون مكوّن من رموز كثيرة.

بيت

بما أن هذه المضامين دقيقة جداً، حاول تجنبها، إلا عند الضرورة القصوى.

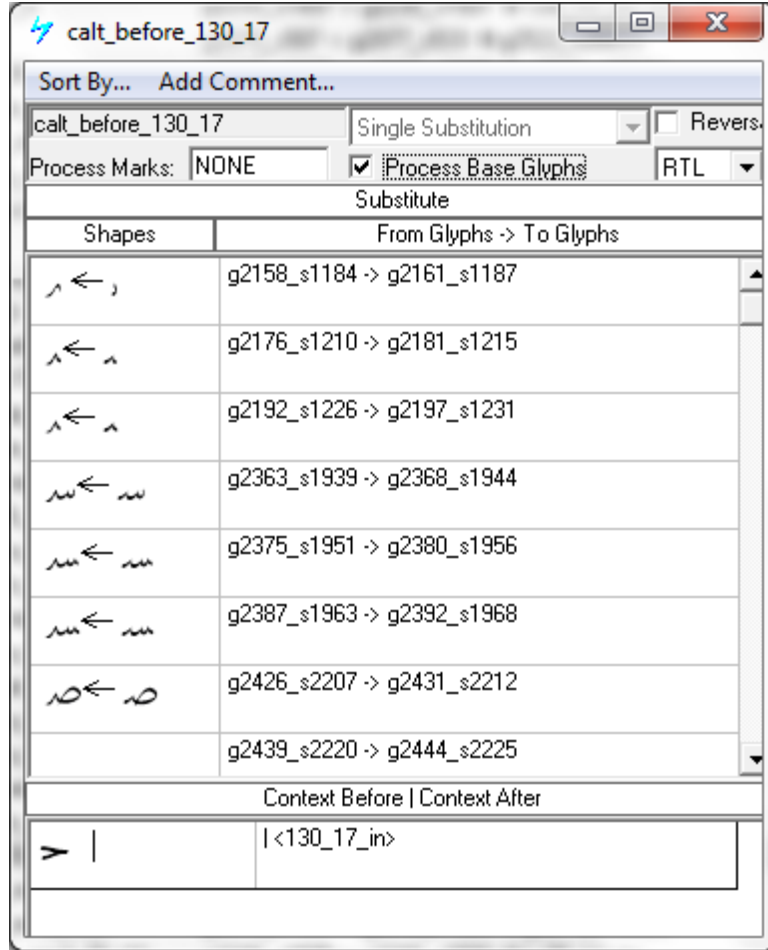
د ا هـ	g2158_1184  g2176_1210
د ا ب	g2158_1184  g2156_1177

الآن وقد وضحت فكرة المضمون (على ما أمل) فلننظر في وسيلة استخدامها مع متغيّرات المضمون.

مضمون «إطاري».

## ٤. ميزة <calt>

<calt> هي مختصر عبارة متغيّرات المضمون (contextual alternates)، وهي الميزة التي تحتاج إليها لخلق البدائل الضرورية لعملتنا. لتطّبق ميزة <calt>، يقرأ محرك النص الميزات التالية وبهذه الترتيبية: أولاً، التفكيك بتطبيق ميزة <ccmp>، ثانياً ميزات التموضع <fina>، <medi>، <init>، <isol>، وثالثاً أحرف المزوجة (ligatures) المطلوبة <rlig>.



لائحة بدائل <calt> في نموذجنا. كما تلاحظ، اختفت جميع الـ Unicode من القائمة الآن. من الضروري أن تلغي كل الـ Unicode في أثناء قراءة محرك النص الأولي (call) للـ font، فحينها تُطّبق ميزات الـ <ccmp>.

تستبدل ميزات التموضع الاساسية أشكال الحروف المنفصلة بالأشكال «النموذجية» التموضعية. اما في ميزة متغيّرات المضمون <calt>، نحتاج إلى أن نستبدل الشكل «النموذجي» التموضعي، برمز الشكل المناسب، طبقاً للمضمون الذي ينبغي للرمز أن يظهر فيه. في نموذجنا الأولي لكلمة «بحر»، نحتاج إلى خلق lookup تابعة لـ <calt>، الظاهرة في الصفحة السابقة.

لنذكر بإيجاز ما تصفه لائحة البحث هذه:

- يشير المضمون إلى أننا عند موضع يسبق الحاء الوسطية «ح».
- يصف البديل ما يجب فعله في هذا المضمون (وضع الشكل المناسب مكان الشكل «النموذجي»).
- في هذا النموذج نعيد هذا الأمر مع كل حرف وشكل تموضعي. كما تلاحظون نستبدل أيضاً أحرف الحاء والسين أو الصاد... وأشكالها الوسطية («ح»، «س»، «ص»).

إن ميزة <calt> قوية جداً، وتتيح بدائل عديدة. ولا بد من أن نتذكر أمراً مهماً جداً، وهو أن الـ lookups تطبق من أول أحرف الكلمة، ثم تطبق تبعاً بحسب نظام ظهورها في قائمة لوائح بدائل voLT، من الأعلى إلى الأسفل. فإذا أنشأت lookups كثيرة، فخذ الحذر الشديد، فحين تطبق lookup ما، تتبدل الرموز، ولذا فلا بد لكل اللوائح التالية من أن تأخذ بالحسبان هذا التغيير، في كل الرموز المبدلة. ويمكن أن يصبح هذا معقداً جداً حين تتعامل مع lookups كثيرة، وقد تحتاج إلى وقت لإعادة الأمور إلى نصابها كما يجب. عليك اختبار فونتك بانتظام وأنت تضع لوائح <calt>. فليس نادراً أن يتطلب الفونت المعقد، عشرات لوائح الـ <calt>.

لكن قبل أن تبدأ اختبار فونتك، لا بد من إلقاء نظرة على ميزات أخرى، وهي التي تختص بتموضع الرموز.

## 0. ميزة تموضع الحروف المتصلة (cursive positioning feature)

تعيين تموضع الرموز (glyphs) في الفونتيات المسطرية البسيطة ليس مشكلة، ما دامت لجميع الرموز وصلات أفقية تماماً. ومن أجل تعيين موقع الرمز ما عليك إلا أن تضعها واحداً تلو الأخرى على الخط الأفقي المستقيم.

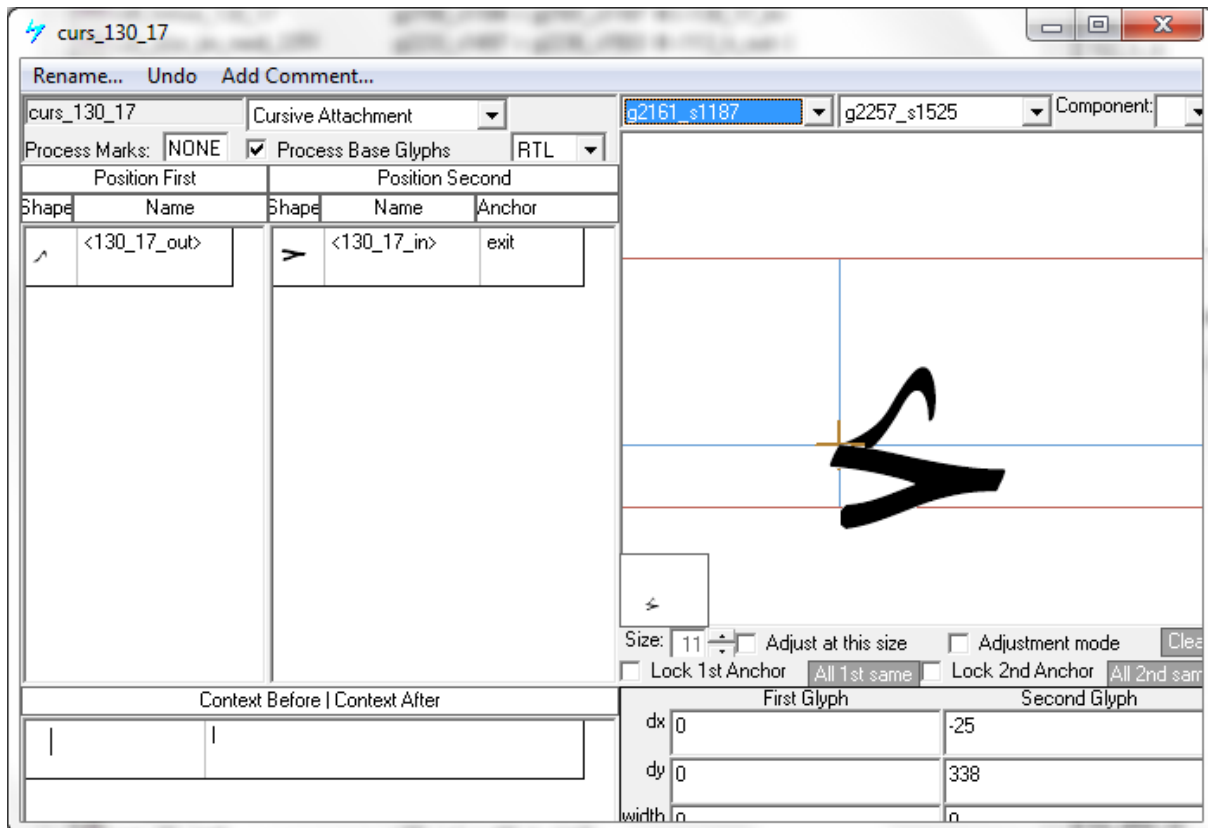
لكننا إذا نظرنا في نوع الوصلات المطلوبة في نموذجنا:



نلاحظ أن هذه الوصلات تفترض وضع كل من الرموز في موضعه الصحيح تماماً.

من أجل هذا، نحتاج إلى أن نستخدم ميزة جديدة اسمها < curs>، (لتعيين تموضع الحروف المتصلة cursive positioning). إنها ميزة لتعيين التموضع، تستخدم نوعاً خاصاً من الرابط (يسمى رابط الأحرف المتصلة cursive attachment)، لتعيين أسلوب وضع الرموز في مواضعها.

ولا بد من أن تُلمَّ بأسلوب استخدام التثبيت أو المرتكز التموضعي (Anchor Positioning)، للاستفادة من هذه الميزة، وإلا يرجى أن تعود إلى ملفنا التعليمي الأول. ولن أدخل هنا في تفاصيل الـ «lookup» هذه.



لائحة <curs> تقليدية تصف الموقع النسبي للرموز. تُستخدم ميزة <curs> الوحدات الداخلية في الفونت (الـ EM)، فلا حاجة إذن إلى معلومات «قياسية» أخرى.

وعليك وأنت تستخدم هذه الأنواع من الوصلات، أن تتذكر أمراً مهماً، هو أن الفونت لم يعد يرسو على سطر أفقي واحد. وقد تنشأ من ذلك مشكلات، في المسافة بين أسطر النص (leading). فمثلاً إذا اتخذنا نموذجاً هذه الكلمة التي لا معنى لها: ...

فونت

في هذه الحال، لا بد من أن تكون المسافة سطر النص والآخر عريضة جداً. ولا تستطيع بعض محركات النص أن تستخدم هذه الميزة كما يجب، ولربما عليك أن تستعين بمحرك نص متطور (مثل «أدوبي إنديزاين» نسخة شرق الأوسط InDesign ME) من أجل الحصول على أفضل ما يمكن من فونتك.

## خلاصة

مستعيناً بهذه الميزات الجديدة في تصميم الفونت، يمكنك الآن أن توسّع الإمكانيات المتاحة لك، وأن تصمم فونتاتاً متطورة جداً. لقد أغفلتُ في هذا الملف التعليمي ميزة مهمة أخرى لتعيين نوع آخر من التوضع اسمها Kerning. تمكّنك هذه الميزة <kern>، من تعيين مواضع الرموز غير المتصلة، جنباً إلى جنب (أي التحكم بحيز الفراغ بين الحرف والآخر). فمثلاً قد ترغب في الحصول على تموضع جميل للكلمة «دعيل»، كما يلي:

دعيل

لبلوغ هذا المستوى من التعقيد والتطور، ستحتاج إلى خلق عدد كبير من من لوائح ال kerning، ومضامينها، وهكذا. ولست متأكداً من أن OpenType له القوة الكافية مما يجعله يعالج كل هذه الحالات كما ينبغي. لكن هذا موضوع آخر. الأکید أنك إذا أردت أن تستكشف هذه الأمور، فإنك ستواجه الحدود التي تحد عمل تقنية OpenType. وقد تحتاج إلى استخدام أدوات عمل أقوى من VOLT، وقد يكون عليك أن تخلق أدوات عملك بنفسك، باستخدام «حقيبة تطوير فونت من أدوبي» (Adobe Font Development Kit (AFDKO)، للـ OpenType المتاح على:

<https://www.adobe.com/devnet/opentype/afdko.html>

حين نبدأ باستخدام AFDKO، نكون قد غادرنا موضوع تطوير الفونت، ودخلنا في دنيا تطوير البرامج الحاسوبية، وهذا يخرج عن نطاق اختصاص هذا الملف التعليمي. في أي حال، حين تستخدم الميزات التي وصفناها في هذا الملف، يمكنك أن تخلق فونتات قوية ومهمة جداً، لكن تذكر أنك حين تبدأ باستخدام رموز متغيرة (alternates) كثيرة، يصبح تعيين تموضع النقاط أعقد شيئاً فشيئاً. وقد تحتاج أحياناً إلى استخدام ميزة المضمون التموضعي للنقاط (contextual dot positioning) من أجل الحصول على أفضل موقع للنقطة، في مضامين متنوعة.

# وضع العقدة في المنشاد

كذلك، يعد إتمام خلق فونت عملاً يقتضي جهداً كبيراً، ووقتاً طويلاً، ومهارة عالية، لكن النتيجة عندئذ تكون مجزية.

## معجم المصطلحات

contextual =	مضمون
alternates =	متغيّرات
grapheme, base shape =	متن الحرف
stylistic alternates =	متغيّرات الأسلوب
contextual alternates =	متغيّرات المضمون
decomposition =	تفكيك
composition =	تكوين
anchor positioning =	التثبيت أو المرتكز التموضعي

